UNIVERSITY OF CAMBRIDGE

Statistical methods for multi-omic data integration

Author
Alessandra Cabassi

Supervisor
Dr Paul DW Kirk

This thesis is submitted for the degree of
Doctor of Philosophy

in the

School of Clinical Medicine
University of Cambridge

St Catharine's College

September 2020

To Cristina, Maria Laura, Michele, and Francesco

Declaration of Authorship

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the Clinical Medicine Degree Committee.

Alessandra Cabassi
September 2020

Abstract

Statistical methods for multi-omic data integration

Alessandra Cabassi

The thesis is focused on the development of new ways to integrate multiple 'omic datasets in the context of precision medicine. This type of analyses have the potential to help researchers deepen their understanding of biological mechanisms underlying disease. However, integrative studies pose several challenges, due to the typically widely differing characteristics of the 'omic layers in terms of number of predictors, type of data, and level of noise.

In this work, we first tackle the problem of performing variable selection and building supervised models, while integrating multiple 'omic datasets of different type. It has been recently shown that applying classical logistic regression with elastic-net penalty to these datasets can lead to poor results. Therefore, we suggest a two-step approach to multi-omic logistic regression in which variable selection is performed on each layer separately and a predictive model is subsequently built on the ensemble of the selected variables.

In the unsupervised setting, we first examine *cluster of clusters analysis* (COCA), an integrative clustering approach that combines information from multiple data sources. COCA has been widely applied in the context of tumour subtyping, but its properties have never been systematically explored before, and its robustness to the inclusion of noisy datasets is unclear. Then, we propose a new statistical method for the unsupervised integration of multi-omic data, called *kernel learning integrative clustering* (KLIC). This approach is based on the idea to frame the challenge of combining clustering structures as a multiple kernel learning problem, in which different datasets each provide a weighted contribution to the final clustering.

Finally, we build upon the notion of the posterior similarity matrix (PSM) in order to suggest new approaches for summarising the output of MCMC algorithms for Bayesian mixture models. A key contribution of our work is the observation that PSMs can be used to define probabilistically-motivated kernel matrices that capture the clustering structure present in the data. This observation enables us to employ a range of kernel methods to obtain summary clusterings, and, if we have multiple PSMs, use standard methods for combining kernels in order to perform integrative clustering. We also show that one can embed PSMs within predictive kernel models in order to perform outcome-guided clustering.

Acknowledgements

First and foremost, I would like to express my sincere gratitude towards my supervisor, Dr Paul DW Kirk, for his unfailing enthusiasm and patience and for offering me great guidance and advice throughout the duration of my PhD. I am also grateful to my advisor Prof Sylvia Richardson for her support and encouragement.

Besides, I would like to thank our collaborators at the Department of Hæmatology, Dr Denis Seyres and Prof Mattia Frontini, for the fruitful exchanges over the course of my PhD and for letting me use their data in this thesis, and the SOMX and PREM research groups, for the valuable discussions throughout the years.

Furthermore, I would like to acknowledge my examiners Dr Simon Rogers and Prof Pietro Liò for their insightful comments and helpful suggestions.

Finally, I sincerely thank my friends and family for their continuous and unparalleled love, help and support.

Preface

I made a stylistic choice to use the first-person plural throughout my thesis. However, this thesis is the result of my own work, except as specified below and in the text.

The work presented in Chapter 2 was carried out as part of a large collaboration with multiple departments. I performed all the analyses presented in this thesis, except for the data preprocessing, which was done by Dr Denis Seyres, and the analysis of the Fenland cohort, that was conducted by Dr Maik Pietzner. The content of Chapter 2 has been included in two preprints:

- Seyres, Denis, Alessandra Cabassi, et al. (2020). "Transcriptional, epigenetic and metabolic signatures in cardiometabolic syndrome defined by extreme phenotype". *bioRχiv preprint*, 10.1101/2020.03.06.961805.
- Cabassi, Alessandra et al. (2020). "Two-step penalised logistic regression for multi-omic data with an application to cardiometabolic syndrome". *arχiv preprint*, 2008.00235.

The latter is currently under review at the scientific journal *Statistical Applications in Genetics and Molecular Biology*.

The content of Chapter 3 has been published in *Bioinformatics*; two accompanying software packages have been made available on The Comprehensive R Archive Network:

- Cabassi, Alessandra and Paul DW Kirk (2020). "Multiple kernel learning for integrative consensus clustering of 'omic datasets" *Bioinformatics*, btaa593.
- Cabassi, Alessandra, Mehmet Gönen, and Paul DW Kirk (2020). "klic: Kernel Learning Integrative Clustering". *R package* klic.
- Cabassi, Alessandra and Paul DW Kirk (2020). "coca: Cluster-Of-Clusters Analysis". *R package* coca.

The content of Chapter 4 has been made available online as a preprint:

- Cabassi, Alessandra, Sylvia Richardson, and Paul DW Kirk (2020). "Kernel learning approaches for summarising and combining posterior similarity matrices" *arχiv preprint*, 2009.12852.

CONTENTS

LIST OF FIGURES

B Appendix to Chapter 3

C Appendix to Chapter 4

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABBREVIATIONS

ADIPO-IR adipose tissue insuline resistance

ALT alanine amino-transferase

AML acute myelogenous leukemia

ANF affinity network fusion

ARI adjusted Rand index

AST aspartate amino-transferase

BCC Bayesian consensus clustering

BHC Bayesian hierarchical clustering

BLCA bladder urothelial adenocarcinoma

BMI body mass index

BP biological process

BRCA breast cancer

CC consensus clustering

CDF cumulative distribution function

CE cell component

ChIP chromatin immunoprecipitation

CIMLR cancer integration via multikernel learning

CMS cardiometabolic syndrome

COAD colon adenocarcinoma

COCA cluster-of-clusters analysis

CpG 5′ – C – p – G – 3′

CRAN Comprehensive R archive network

CV cross-validation

CVD cardiovascular disease

DBN deep belief network

DP Dirichlet process

DR dimensionality reduction

EM expectation-maximisation

EN elastic-net

EPSGO efficient parameter selection via global optimisation

FDR false discovery rate

FFA free fatty acid

FWER family-wise error rate

GBM glioblastoma multiforme

GIMM Gaussian infinite mixture model

GLC glucose

GOTO gene ontology term overlap

GP Gaussian process

GWAS genome-wide association study

HDL-C high-density lipoprotein cholesterol

HNSC head and neck squamous cell adenocarcinoma

HOMA-IR homeostasis model assessment of insulin resistance

HOMA2-IR homeostasis model assessment of insulin resistance

hsCRP high sensitivity C-reactive protein

i.i.d. independent and identically distributed

iPF integrative phenotyping framework

IPF-LASSO integrative LASSO with penalty factors

KIRC renal cell carcinoma

KLIC kernel learning integrative clustering

LAR leptin to adiponectin ratio

LASSO least absolute shrinkage and selection operator

LDL-C low-density lipoprotein cholesterol

LPP locality preserving projections

LRAcluster low-rank approximation-based multi-omics data clustering

LUAD lung adenocarcinoma

LUSC lung squamous cell carcinoma

MCMC Markov chain Monte Carlo

MDI multiple dataset integration

MF molecular function

mice multivariate imputation by chained equations

miRNA microRNA

MKL multiple kernel learning

ML machine learning

MM mixture model

MOC matrix of clusters

MONET multi-omic clustering by non-exhaustive types

MR misclassification rate

mRNA messenger RNA

MTP-EN multi-tuning parameter elastic-net

NASH non-alcoholic steatohepatitis

NEMO neighbourhood-based multi-omics clustering

OG outcome-guided

OV serous ovarian carcinoma

PAC proportion of ambiguous clustering

PAM partitioning around medoids

PathMe pathway based multi-modal sparse autoencoders for clustering of patient-level multi-omics data

PCA principal component analysis

PEAR posterior expected Rand index

PINS perturbation clustering for data integration and disease subtyping

PSD positive semi-definite

PSM posterior similarity matrix

QP quadratic programming

RBF radial basis function

READ rectal adenocarcinoma

RNA-seq RNA sequencing

SE standard error

SIMLR single-cell interpretation via multi-kernel learning

SNF similarity network fusion

SVM support vector machine

TC total cholesterol

TCGA The Cancer Genome Atlas

TF transcription factor

TG triglycerides

TRIPOD transparent reporting of a multivariable prediction model for individual prognosis or diagnosis

UCEC endometrial cancer

VAE variational autoencoder

WP10 work package 10

1

INTRODUCTION

High-throughput technologies have made it possible to collect an enormous amount of *'omic* data in a variety of contexts, ranging from medical (e.g. oncology; Aure et al., 2017) to commercial applications (Zhu et al., 2012). To give an example, data for over 10,000 cancer patients have been collected for The Cancer Genome Atlas (TCGA) project alone (The Cancer Genome Atlas Research Network et al., 2013) with the goal of uncovering similarities and dissimilarities between different tumour types and exploit them to develop more specific and effective cancer treatments.

Common to multi-omic studies is the desire to deepen biological insight, to improve our ability to characterise biological processes, and/or to enable better predictions (e.g. regarding disease outcome) to be made. In precision medicine, for example, the goal is to exploit 'omic data in order to increase the accuracy and efficiency of medical decisions and treatments (Hasin, Seldin, and Lusis, 2017). Multi-omic studies have been extensively used, for example, to discover novel disease subtypes in cancer (see e.g. Shen et al., 2012 for glioblastoma; The Cancer Genome Atlas Research Network, 2012 for breast cancer; The Cancer Genome Atlas Research Network, 2017 for liver cancer), inflammatory myopathies (Greenberg et al., 2002), and auto-immune diseases (Lee et al., 2011).

The challenge is then to develop statistical methods that can cope with the huge amount and variety of data and, at the same time, are able to take into account the fact that each data *layer* has different properties and characteristics. In this thesis, novel methodologies are proposed to address some of the open questions in this field. In particular, we focus on supervised and unsupervised integration of multi-omic datasets.

Chapter outline

The motivation for developing statistical methodologies for multi-omic data integration is elucidated in Section 1.1. The 'omic data types considered in this thesis are introduced in Section 1.2. Then, the statistical setting and mathematical notation used throughout this thesis are introduced in Section 1.3. The goals of supervised and unsupervised integration of multi-omic data are presented in Section 1.4, followed by a description of some of the issues associated with the statistical

analysis of multi-omic data, in Section 1.5. Finally, we give an overview of the structure of the thesis in Section 1.7.

1.1 MOTIVATION

A molecular term followed by the suffix *'omics* indicates a comprehensive or full assessment of a set of molecules (Hasin, Seldin, and Lusis, 2017). Before introducing the types of 'omic data layers used in this thesis, we explain why it is important that these datasets are collected and analysed together.

The most mature of the 'omic fields is *genomics*. As opposed to *genetics*, which focuses on single genes or variants, the word genomics indicates the study of an organism's complete set of DNA, the *genome* (*WHO definitions of genetics and genomics*). In the early 2000s, thanks to the availability of large amounts of whole-sequencing data, genome-wide association studies (GWAS) were designed to identify genotype-phenotype associations (Klein et al., 2005). These studies have made it possible to improve understanding of many complex traits, including both common and rare diseases (Calvo et al., 2006). For instance, they have led to the discovery of heritable mutations linked to breast cancer, such as mutations in the BRCA1 and BRCA2 genes (Venkitaraman, 2014). Nevertheless, it has become clear in recent years that the genetic variants identified by GWAS only account for a fraction of the heritability of specific traits and that gene regulation also plays an important role in explaining phenotype (Visscher et al., 2012). Moreover, the same genetic variants can lead to different outcomes depending on external factors (Tam et al., 2019). Finally, while the analysis of the genome can only reveal correlations, by integrating multiple 'omic data layers, one might expect to uncover potential causes of a certain disease, deepen their understanding of the mechanisms underlying its development, and/or identify putative treatment targets (Hasin, Seldin, and Lusis, 2017). Therefore, one of the reasons why multi-omic analyses have become so widespread, is that they overcome some of the limitations of GWAS.

The central dogma of molecular biology, as formulated by Francis Crick in the 1950s, states that the flow of genetic information is unidirectional: DNA is copied to RNA via a process called *transcription*, and proteins are synthesised from the information contained in the RNA (more precisely *messenger RNA*, as explained below) through a process known as *translation* (Crick, 1958). Since the central dogma was first introduced, the understanding of how genetic information is transferred within biological systems has greatly improved and more complex interactions between their components have been discovered (Shapiro, 2009).

Multi-omic studies take a *holistic* (or *systems*) approach and incorporate different 'omic data layers into coherent models, acknowledging that complex interconnections exist between them (Huang, Chaudhary, and Garmire, 2017). This

allows researchers to overcome the limitations of single 'omic analyses and has many advantages. First, each 'omic layer provides different types of information. This can be beneficial in prediction and subtyping analyses such as those presented in this thesis, given that each layer provides complementary information that can be used to produce the final output (Bersanelli et al., 2016). Moreover, when the relative importance of each layer for the problem at hand is unknown, this can be inferred using multi-omic statistical methods. Additionally, if the data collected for one of the statistical units in the study is noisy, information can be borrowed from other layers. Another advantage is that jointly analysing different 'omic layers can provide deeper insights into the mechanisms underlying the biological phenomenon of interest compared to single-omic analyses, potentially uncovering interactions between the 'omic layers (Rogers et al., 2008). In Figure 1.1 is depicted a simplified representation of the main 'omic types considered in this thesis and some of the ways in which they are related.

Holistic approaches combining multiple 'omic data layers have proved useful in several fields. They have been widely applied, for instance, to cancer studies (Karczewski and Snyder, 2018), with various goals, including: locating genetic mutations that are drivers for cancer development The Cancer Genome Atlas Research Network (2017), identifying molecular signatures of specific tumour types (Kristensen et al., 2012), discovering cancer subtypes (The Cancer Genome Atlas Research Network, 2011; Sato et al., 2013). Recent progress in this field, from the point of view of the statistical methodology, has been reviewed and summarised, among others, by Kristensen et al. (2014) and Nicora et al. (2020).

1.2 MULTI-OMIC DATA

In this section we give a brief description of the types of datasets that are considered in this thesis.

Genomics

As we have mentioned, genomics is concerned with the study of an organism's full DNA, and the advent of high-throughput DNA sequencing technologies generated a proliferation of whole-genome studies over the past two decades (Goodwin, McPherson, and McCombie, 2016). The genome-wide associations studies mentioned above focus on finding specific genetic variants associated with the trait of interest. However, other features of the genome can explain phenotype variability. For instance, in many living organisms, some sections of the genome are repeated. In humans, two-thirds of the genome are composed of repeats (Koning et al., 2011). Therefore, copy number variation is also routinely recorded and examined in genomics. Data of this type have been used to gain insight into human diseases (McCarroll and Altshuler, 2007) such as autism (Sebat et al., 2007)

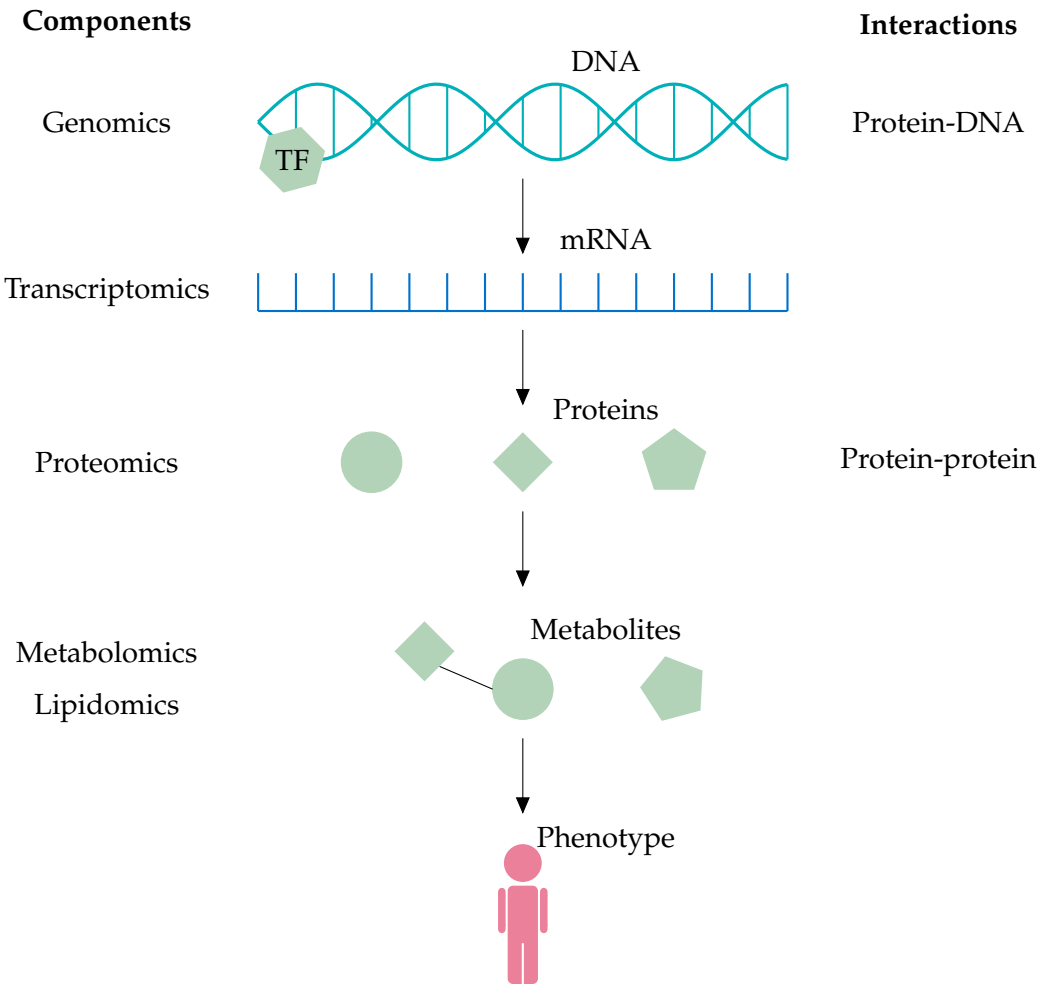


FIGURE 1.1: Simplified representation of the interactions between some of the 'omic data layers considered in this thesis. DNA is transcribed to make different types of RNA. For simplicity, only mRNA is represented here. Proteins are synthesised from the information contained in the mRNA. The metabolites present in a cell are determined by the proteins as well as environmental factors. Proteins interact not only between them, but also with the DNA. For example, transcription factors (TFs) are proteins that initiate and regulate the transcription of genes. Figure freely adapted from Joyce and Palsson (2006).

and schizophrenia (St Clair, 2009).

Transcriptomics

The *transcriptome* is the set of all RNA transcripts produced under specific circumstances or in a specific cell (Watson et al., 2014, Chapter 7). The field of *transcriptomics* therefore examines genome-wide RNA levels. The technique that is used most frequently to study the transcriptome is RNA sequencing (RNA-seq; Allison et al., 2006; Ozsolak and Milos, 2011). This sequencing technique is used to reveal the presence and quantity of RNA in a biological sample. Different types of RNA exist. In most applications, interest is focused on *messenger RNA* (mRNA), which is the one containing the necessary information for protein synthesis. In one of the real data applications considered in this thesis, however, we also have *microRNA* expression data available. microRNAs, commonly referred to as miRNAs, are small non-coding RNA molecules that have been identified more recently than mRNA (Bartel, 2004). They regulate gene expression by influencing the stability of mRNA or its ability of being translated (Watson et al., 2014, Chapter 8).

Proteomics

The transcription rate of a gene only gives an approximate estimate of the expression of the protein encoded by it. This is because the corresponding mRNA can decay quickly or not be translated much and therefore looking only at gene expression can be misleading. This is why it is important to also identify and quantify the cellular levels of each protein encoded by the genome, the so-called *proteome*, in the cell or tissue of interest. This is what is commonly referred to as *proteomics* (Joyce and Palsson, 2006). Protein-protein interactions and post-translational modifications are also of interest, as they influence the proteins' activity (Watson et al., 2014, Chapter 7).

Metabolomics

The term *metabolite* indicates any intermediate or end product of metabolism (Venes, 2017). *Metabolomics* is then the systematic study of metabolites within a specific cell or tissue (Tweeddale, Notley-McRobb, and Ferenci, 1998). As pointed out by Hollywood, Brison, and Goodacre (2006), metabolomics is complementary to transcriptomics and proteomics, because multiple factors (i.e. post-transcriptional and post-translational events) influence metabolic fluxes. Thus, the metabolome is considered to be *closer* to the phenotype (see Figure 1.1).

Lipidomics

A type of metabolite that is of particular interest for one of the motivating examples of this thesis are *lipids*. For this reason, we also introduce *lipidomics*: the system-level analysis of lipids and their interactions with proteins and with other

metabolites (Wenk, 2005). A number of human and animal diseases have been associated with abnormal lipid levels, including cancer (Reynolds, Maurer, and Kolesnick, 2004), diabetes (Shi and Burn, 2004), and neurodegenerative diseases (Cutler et al., 2004).

Epigenomics

Epigenomics is the study of the complete set of epigenetic modifications on the genetic material of a cell, known as the *epigenome* (Fazzari and Greally, 2004). Changes to the epigenome are heritable and can result in modifications of chromatin structure and to the function of the genome (Yong, Hsu, and Chen, 2016).

In this thesis we analyse both histone modifications and transcription factor data. In eukaryotic cells, histones are the proteins around which the DNA is packed (Campos and Reinberg, 2009). Histone modifications include a few chemical modifications, each having a different effect on gene regulation (Kuo and Allis, 1998). Here we only consider DNA methylation, which occurs when methyl groups are added to a DNA molecule (Moore, Le, and Fan, 2013). If a gene promoter is methylated, transcription of the corresponding gene is usually repressed (Joyce and Palsson, 2006).

Transcription factors (TFs) are proteins that control the transcription rate of certain genes by binding to specific DNA binding sites (Latchman, 1997). The data that we analyse in this thesis have been collected using either chromatin immunoprecipitation DNA sequencing (ChIP-seq) or ChIP-chip techniques (i.e. where chromatin immunoprecipitation experiments are performed on DNA microarrays, also known as *chips*) to identify the binding sites of each transcription factor of interest.

Other 'omics

Other 'omic data types that are not considered here include, among others: *microbiomics*, the study of all microorganisms of a given community (see e.g. Egert et al., 2006), *glycomics*, the comprehensive study of glycan structures of a given cell type or organism (Wandall et al., 2017).

1.2.1 *Individual-level versus gene-level multi-omic analyses*

We have already mentioned a few multi-omic studies where the goal is to uncover the structure of a population of individuals. For example, in precision medicine applications such as those cited above, multiple types of 'omic data are combined to characterise the molecular signature of the disease of interest, investigate disease pathogenesis pathways and develop personalised therapies.

However, it is important to note that multi-omic datasets can also be collected for a specific set of genes; datasets of this kind can be used to identify sets of genes that have similar functions. A typical example is *transcriptional module discovery* (Wu et al., 2002): genes belonging to the same transcriptional modules are regulated by the same transcription factors and share the same biological function. Combining gene expression and transcription factor binding data can help identify transcriptional modules and improve the understanding of cellular processes (Savage et al., 2010).

Throughout this thesis, we consider and develop methodologies that can be applied to both types of data. In fact, two out of three real data applications considered in this work belong to the first category (i.e. individual-level multi-omics), while the remaining one is part of the second category (i.e. gene-level multi-omics).

1.3 MATHEMATICAL NOTATION

In this section, we set out the mathematical framework and notation of the methodological and applied problems tackled in this thesis.

In every application considered in this thesis, we have a collection of N statistical units (e.g. individuals or genes) for which M different types of 'omic measurements have been taken. We indicate each 'omic data *layer* by X_m and by P_m the number of features in that layer, that is $X_m \in \mathbb{R}^{N \times P_m}$ for $m = 1, \dots, M$. Moreover, throughout this thesis we indicate by X the matrix containing all data layers, defined as

$$X = [X_1, \dots, X_M] \in \mathbb{R}^{N \times P},$$

where $P = P_1 + \cdots + P_M$ is the total number of features available for each individual, taking all data layers together. In supervised settings we also have a response y_n for each observation $n = 1, \dots, N$ (Figure 1.2).

1.4 STATISTICAL METHODS FOR MULTI-OMIC INTEGRATION

There are many ways in which multi-omic analysis approaches could be classified. Subramanian et al. (2020), for instance, grouped them by the type of biological question they address. Here, we divide them into supervised and unsupervised analyses. Both types of analyses are briefly introduced here; however, to avoid repetition, we expand upon these concepts in the relevant thesis chapters only.

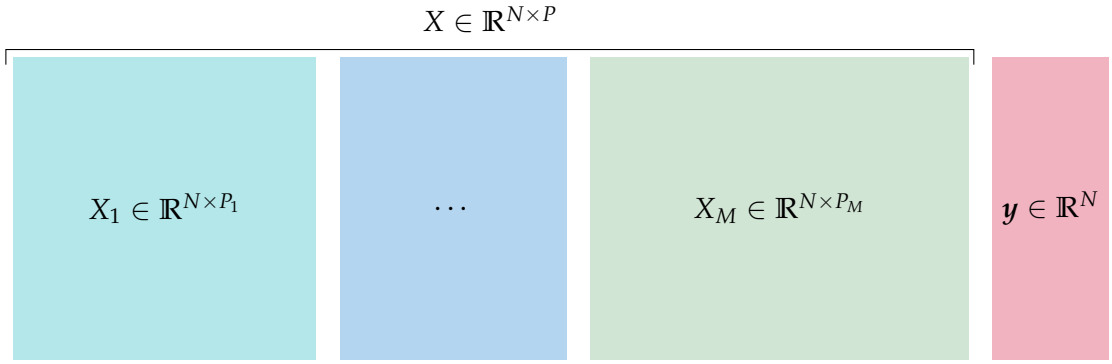


FIGURE 1.2: Schematic representation of multi-omic data. $X_1, X_2, \ldots, X_M$ are 'omic layers relative to the same observations, $\boldsymbol{y}$ is a response/outcome variable. The data matrix comprising all data layers is indicated by X. Each colour indicates a different 'omic layer.

Supervised integration

In supervised settings, the 'omic data layers are commonly used to learn a predictive model for the response variable of interest. In precision medicine applications, for instance, this can be a binary variable indicating the disease status of each individual, a continuous score that is known to be associated to the severity of disease, or survival/relapse time (see e.g. Seoane et al., 2014; Zhao et al., 2015).

Additionally, the supervised integration of multiple 'omic data layers can also be exploited for biomarker discovery via variable selection methods (Rohart et al., 2017). This can give insights into the molecular mechanisms underlying the pathogenesis and ætiology of a certain disease.

To give an example, one of the perspectives of development for the supervised integration of multi-omic data identified very recently by Karczewski and Snyder (2018) was the construction of predictive models of disease risk, to be used to perform early detection of disease on healthy individuals. Karczewski and Snyder envisioned, for example, the creation of models of cardiovascular disease that could stratify individuals into high- and low-risk categories, combining genomic and metabolomic data. The supervised integration problem considered in this thesis is closely related to this.

In this thesis we focus on binary classification models. In particular, we consider penalised likelihood approaches, which are scalable, widely used, and have the advantage of having efficient implementations available. A more detailed explanation of how these methods work is presented in Chapter 2.

Unsupervised integration

The unsupervised integration methods considered in this thesis are integrative clustering algorithms. The goal of multi-omic integrative clustering is to identify individuals or genes that appear similar across most of the available data layers. For example, Hoadley et al. (2014) used cancer samples from 12 different tumour tissues to identify cancer subtypes having similar molecular profiles. The novel cancer subtypes found by Hoadley *et al.* were shown to provide prognostic information, which could be used to develop targeted therapy for each subtype. The unsupervised methods developed in this thesis target applications of this kind. An extensive literature review of unsupervised methods for the integration of multiple 'omic data layers is presented in Chapter 3.

It is important to note, however, that unsupervised integration methods of multi-omic data also include factor analysis (Argelaguet et al., 2018) and variance decomposition methods (Lock et al., 2013). The goal of this type of methods is to identify a set of latent factors which capture the different sources of variability across 'omic data layers. We do not focus on these methods here, since our objectives are substantially different, as is explained in Chapter 3.

1.5 STATISTICAL CHALLENGES OF MULTI-OMIC INTEGRATION

Detailed reviews of current statistical methods for integrative 'omics include those of Richardson, Tseng, and Sun (2016), Bersanelli et al. (2016), Huang, Chaudhary, and Garmire (2017), Li, Wu, and Ngom (2018), and Subramanian et al. (2020). We summarise here the most well-known issues associated with the joint statistical analysis of multiple 'omic datasets that have emerged over the years.

Different types of data

'Omic data layers can be of different types. For instance, measurements expression level can either be continuous or count data. The transcription factor data analysed in the following chapters, instead, are binarised so that ones indicate interactions between a DNA-binding site and a transcriptional regulator, zeros represent absence of interaction. When developing integrative methods for multi-omic data, it is crucial to take this into account.

Different layer sizes

The number of features available can differ greatly across data layers. For humans, for example, genomic data layers can have a huge number of features, since their DNA contains approximately 3 billion base pairs and more than 20,000 genes (Watson et al., 2014, Chapter 2). Human metabolomic and lipidomic data, on the contrary, are usually of much lower dimension (Wishart, Tzur, and Knox,

2007). When all data layers are considered together, untangling the contributions of each layer is often difficult. Moreover, there is a risk of inadvertently letting larger datasets have greater impact on the analysis than smaller ones, even though they are not necessarily the most informative ones.

Varying levels of noise

High-throughput techniques are known to be plagued by high levels of technical noise (see e.g. Marshall, 2004; Ioannidis, 2005) and batch effects (Leek et al., 2010). Similar to the previous point, this one too must be tackled by carefully monitoring how much each data layer contributes to the final output, so that noisy datasets are prevented from having excessive influence on the analysis.

High computational cost

Due to the ever increasing size of high-throughput datasets, multi-omic data usually comprise very large numbers of features P and, in some instances, also present a large number of observations N. This can make it impossible or extremely computationally demanding to apply existing methods to these types of data. As we discuss at length in the next chapters, possible solutions to this problem include reducing the number of features with variable selection methods or other pre-screening techniques, and/or making use of two-step approaches where the dimensionality of each layer is reduced in the first step independently from the other layers. However, care must be exercised when using these strategies, as these may result in unclear or unwanted up- or down-weighting of the impact of each dataset on the final prediction/clustering.

Large P small N

Another issue related to the large size of high-throughput datasets is the so-called *large P small N* problem. For instance, supervised methods are known to require extra care when the number of variables P exceeds the number of statistical units N. This is because classical statistical methods designed for large N small P situations may fail or overfit the data (James et al., 2013). More details about this issue are given in Chapter 2, where penalised likelihood methods are introduced. Moreover, in situations where one may want to perform an hypothesis test for each 'omic variable, multiple testing correction makes the overall procedure extremely conservative (McIntyre et al., 2000), as we shall see in Chapter 2. Combining multiple large P, small N datasets of different types raises new questions, as the large P small N problem is exacerbated by the presence of multiple datasets.

1.6 TWO-STEP INTEGRATIVE METHODS

Integrative methods can also be divided into *joint* and *two-step* models. In the former, all 'omic layers are analysed together. In two-step approaches, instead, dimension reduction is first performed on each 'omic layer separately. All 'omic layers are then integrated in a subsequent step to generate the final output. These methods have sometimes also been referred to as *sequential analysis* (Kristensen et al., 2014) or *late integration* methods (Rappoport and Shamir, 2018).

While joint models can in principle detect and exploit dependencies between features in different layers, they suffer from the issues described in the previous section and, for that reason, are often not applicable to large multi-omic datasets in practice. Therefore in this thesis we focus our attention on two-step integrative methods, which overcome these issues in a principled way. As we shall see, the first step enables us to solve the problems of high computational cost and large P small N, by decreasing the dimension of each layer separately. Moreover, in the first step, different types of data can be analysed with different techniques. Therefore, depending on the data type of each layer, the most appropriate method can be selected. Similarly, the problem of unbalanced feature numbers in the 'omic layers can be alleviated by either making sure that a reasonable number of features is selected in each layer or summarising the information of each 'omic layer into more easily comparable statistical objects. Finally, we show in later chapters how the use of appropriate weighting strategies can help tackle the issue of varying levels of noise.

1.7 THESIS OVERVIEW

Chapter 2: Two-step penalised logistic regression for multi-omic data

In Chapter 2 we build two classification models that predict a binary class label on the basis of multiple 'omic layers. We use a wide range of simulation studies to show that these methods identify higher numbers of relevant features and achieve comparable misclassification rates to competitor methods.

Furthermore, we analyse a novel dataset comprising eight layers of 'omic data, as well as a wide range of clinical covariates associated to cardiometabolic disease, for lipodystrophy patients, obese individuals, and blood donors. We describe a multi-omic analysis aimed at identifying molecular signatures of cardiometabolic syndrome and building a predictive model for the risk of developing cardiometabolic syndrome.

The work presented in Chapter 2 was carried out as part of a larger study, that

was done in collaboration with the Department of Hæmatology of the University of Cambridge. Part of the data analysis included in the chapter has been incorporated into a manuscript entitled "Transcriptional, epigenetic and metabolic signatures in cardiometabolic syndrome defined by extreme phenotypes" that is available on bioRχiv (Seyres et al., 2020). The methodological content of the chapter has been published as an arχiv preprint with title "Two-step penalised logistic regression for multi-omic data with an application to cardiometabolic syndrome" (Cabassi et al., 2020), which is currently under review at the *Statistical Applications in Genetics and Molecular Biology* journal[1].

Chapter 3: Multiple kernel learning for integrative clustering of 'omic datasets

In Chapter 3 we present a review of the existing clustering methods for multi-omic data and systematically explore one of them, called *cluster-of-clusters analysis* (COCA). This provides the motivation to propose an alternative strategy named *kernel learning integrative clustering* (KLIC). This method is based on the idea that kernels can be built for each 'omic data layer using consensus clustering (Monti et al., 2003) and combined in a principled way via multiple kernel learning. We assess both methods via extensive simulation studies and compare them to competitor methods. Finally, we apply them to two real clustering problems: pan-cancer clustering and transcriptional module discovery.

The content of Chapter 3 has been published as a manuscript entitled "Multiple kernel learning for integrative consensus clustering of 'omic datasets" (Cabassi and Kirk, 2020b). Together with the article, two R packages, called `coca` and `klic` have been published on the Comprehensive R archive network (CRAN; Cabassi and Kirk, 2020a; Cabassi, Gönen, and Kirk, 2020).

Chapter 4: Summarising and combining posterior similarity matrices

In Chapter 4 we propose a new way of summarising the posterior similarity matrices (PSMs) derived from the Markov chain Monte Carlo output of Bayesian model-based clustering, making use of kernel methods. At the same time, we also prove that PSMs are valid kernels, which can be used as input to KLIC. Using Bayesian model-based clustering algorithms instead of heuristic approaches of Chapter 3 has several advantages: not only it represents a more principled way of seeking a partition of the data, but it also provides elegant solutions to some of the difficulties encountered in Chapter 3, such as the choice of the number of clusters and the selection of the variables relevant for clustering. Moreover, we extend the kernel learning integrative clustering methodology to the outcome-guided setting, where a response variable is used to guide the kernel weighting. Finally, we show how the PSM-based unsupervised and outcome-guided

[1] `https://www.degruyter.com/view/journals/sagmb/sagmb-overview.xml`

methods perform in practice, through a set of simulation studies, and apply these methods to the same real data applications presented in Chapter 3.

The work presented in Chapter 4 has been made available on arχiv as a preprint titled "Kernel learning approaches for summarising and combining posterior similarity matrices" (Cabassi, Richardson, and Kirk, 2020).

Chapter 5: Integrative analysis of cardiometabolic disease data

The cardiometabolic syndrome dataset introduced in Chapter 2 is used in Chapter 5 to show how all the methodological developments of this thesis are linked together. In particular, after using the classification models of Chapter 2 to identify the variables that discriminate between healthy individuals and those affected by cardiometabolic syndrome, we are able to apply the unsupervised and outcome-guided methods of Chapter 3 and 4 to this dataset.

Chapter 6: Discussion

Finally, in Chapter 6 are summarised the overall findings of the thesis and are outlined potential future research areas.

2

TWO-STEP PENALISED LOGISTIC REGRESSION FOR MULTI-OMIC DATA

In this chapter we focus on the problem of making predictions for a binary variable using multiple high-dimensional 'omic layers. In our motivating example, we are interested in finding molecular signatures of cardiometabolic syndrome in eight 'omic layers. We propose a two-step approach in which a variable selection step is first performed on each 'omic layer separately, using penalised logistic regression approaches. In the second step, all the selected variables are included in a ridge-penalised logistic regression model, which is fitted to the data. We compare this approach to (i) fitting a logistic regression model with elastic-net penalty on the data matrix formed by concatenating all layers (matrix X in Figure 1.2), (ii) a newly developed integrative method that fits a regression model on all data types together, but assigning different penalty factors to each of them (Zhao and Zucknick, 2020), (iii) selecting features via a univariate approach and then using all the selected variables to fit a ridge-penalised logistic regression model.

We consider a wide range of simulation studies. In each simulation setting there are two data types with varying characteristics (e.g. number of features, percentage of relevant features, etc.), as well as a smaller data type that only contains features that are known to be associated with the outcome of interest, and therefore are not penalised. In 'omic data applications, this corresponds to having two 'omic datasets and a small set of clinical parameters. The simulation studies show that, depending on the goal of the analysis (e.g. estimation of patient risk of developing a disease, multi-omic signature identification, etc.), different integrative methods should be preferred. If the objective is to make accurate predictions and, at the same time, identify the highest possible number of variables that are relevant for the problem at hand, then our approach is the most suitable.

We apply the developed methods to a novel multi-omic dataset, collected with the aim of understanding and identifying a molecular characterisation of cardiometabolic syndrome, that could improve our understanding of this condition and, as a consequence, help develop new treatment strategies. We analyse eight different types of 'omic data from 184 blood donors, as well as 11 obese and 10 lipodystrophy individuals. We use these data to identify putative signatures of

cardiometabolic syndrome and build a predictive model to determine the probability of belonging to the obese or lipodystrophy group. We investigate the impact of the choice of the elastic-net parameter on the estimated probabilities and the variables selected, by comparing to a rank aggregation approach.

Chapter outline

The chapter is structured as follows. Section 2.1 contains an introduction to penalised regression, with particular focus on the models that have been developed specifically for multi-omic data. The novel methods are introduced in Section 2.2 and simulation studies are performed in Section 2.3 to compare the new and existing methods. The dataset available for this study is presented in Section 2.4. The multivariate data analysis is reported in Section 2.5. The comparison to the univariate analysis of the data is presented in Section 2.6. In Section 2.7, the findings of the multivariate analysis are validated using two external cohorts. Finally, in Section 2.8 we summarise the main findings of this work and the challenges encountered when performing this analysis.

2.1 PENALISED REGRESSION FOR MULTI-OMIC DATA

First, we briefly recall the basics of penalised logistic regression in Section 2.1.1. Then, a review of the predictive models of this type that have been used to integrate multiple 'omic datasets is given in Section 2.1.2. Finally, Section 2.1.3 contains the details of the algorithm that is extensively used in the remainder of this chapter to tune the penalty parameters.

2.1.1 *Penalised logistic regression*

In traditional logistic regression settings, one has a data matrix $X \in \mathbb{R}^{N\times P}$ comprising N observations $\mathbf{x}_n$, $n = 1,\dots,N$, for which a set of p variables has been measured, and a set of binary responses $\mathbf{y} = [y_1,\dots,y_N] \in \{0,1\}^N$, one for each observation in X. Logistic regression is a statistical method that can be used to build a model that predicts the probability that the response y_{new} corresponding to a new observation $\mathbf{x}_{\text{new}}$ is equal to one (Cramer, 2002; Hastie, Tibshirani, and Friedman, 2009). This is done via the *logistic function*

$$\Pr(Y = 1|\mathbf{x}) = \frac{e^{\beta_0+\boldsymbol{\beta}\mathbf{x}}}{1+e^{\beta_0+\boldsymbol{\beta}\mathbf{x}}}\,, \tag{2.1}$$

where β_0 and $\boldsymbol{\beta} = [\beta_1,\dots,\beta_p]$ are the so-called *regression coefficients*. In particular, β_0 is the *intercept* of the model and each β_p is the coefficient corresponding to the

pth variable (i.e. column) in X. The regression coefficients can be estimated by maximising the likelihood of observed data, that is by solving the optimisation problem

$$\max_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^P} l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}),$$

where $l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta})$ indicates the log-likelihood of the data given β_0 and $\boldsymbol{\beta}$. By defining $p(\mathbf{x}) = \Pr(Y = 1|\mathbf{x})$, Equation (2.1) can be rewritten as

$$\log\left(\frac{p(\mathbf{x})}{1 - p(\mathbf{x})}\right) = \beta_0 + \boldsymbol{\beta}\mathbf{x}.$$

The expression on the left hand side is called the *log-odds* (Hastie, Tibshirani, and Friedman, 2009, Chapter 4).

In the presence of large numbers of predictors, the estimates of the regression coefficients β_0 and $\boldsymbol{\beta}$ given by solving the optimisation problem above are highly variable (Hastie, Tibshirani, and Friedman, 2009, Chapter 3). To avoid this problem, shrinkage methods are often used. The most popular shrinkage approaches are ridge regression and *least absolute shrinkage and selection operator* (LASSO).

Ridge regression maximises the maximum likelihood subject to a quadratic penalty on the coefficients. This corresponds to solving the optimisation problem

$$\min_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^P} -l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2, \tag{2.2}$$

where $\|\cdot\|_2$ is the l_2 norm, defined as $\|\boldsymbol{\beta}\|_2 = \sum_{p=1}^{P} \beta_j^2$ and λ is a tuning parameter that determines the strength of the penalty (Hoerl and Kennard, 1970). The parameter λ is usually determined by *cross-validation* (CV), so as to minimise some measure of the average prediction error (Kohavi, 1995). More details about CV and the choice of λ are given at the end of this section.

LASSO regression instead has an l_1-penalty:

$$\min_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^P} -l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1,$$

where $\|\cdot\|_1$ is the l_1 norm, defined as $\|\boldsymbol{\beta}\|_1 = \sum_{p=1}^{P} |\beta_j|$. Contrary to ridge regression, LASSO sets some of the regression coefficients to zero, which is equivalent to doing variable selection. This gives rise to more interpretable predictive models, since the predictions are based on fewer variables (Tibshirani, 1996). However, if there is a set of highly correlated variables associated with the response, LASSO tends to select only one of them (Zou and Hastie, 2005). This might not always

be the desired outcome. For instance, in the application presented here, we want to be able to select all the variables that are predictive of patient status.

In what follows, we make extensive use of a penalised logistic regression method that uses a mixture of the l_1 and l_2 penalties, called the *elastic-net* (EN):

$$\min_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^p} -l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}) + \lambda \left[(1-\alpha) \|\boldsymbol{\beta}\|_2 + \alpha \|\boldsymbol{\beta}\|_1 \right] \tag{2.3}$$

Here $\alpha \in [0, 1]$ is the weight assigned to the LASSO penalty, and $1 - \alpha$ is the weight assigned to the ridge penalty (Zou and Hastie, 2005). Therefore, the number of selected variables decreases as α increases (Hastie, Tibshirani, and Friedman, 2009, Chapter 3).

Choice of the tuning parameters

In EN, the parameters that need to be tuned are λ, which represents the strength of the penalty, and α, which defines how much importance is given to the l_1 penalty relative to the l_2 penalty (see Equation 2.3).

As mentioned previously, a popular method to choose the parameter λ is cross-validation (Kohavi, 1995). Briefly, k-fold CV is a resampling method that helps estimating the out-of-sample error of a predictive model. In the case of logistic regression, the out-of-sample error is usually measured as the average number of misclassified observations in the test set. Indicating by $\hat{y}_i$ the predicted value for observation i and by N_{test} the number of observations in the test set, the *misclassification rate* (MR) is

$$\text{MR} = \frac{\sum_{n=1}^{N_{\text{test}}} \mathbb{1}(y_i \neq \hat{y}_i)}{N_{\text{test}}}.$$

In CV, first the data are split into k folds, using one of the folds in turn as the test set and the others as the training set. The out-of-sample error for a certain value of the parameter of interest is then estimated as the average validation error over the k repetitions. This procedure is repeated for a set of plausible values of the parameter of interest; the value that minimises the average error is selected. Alternatively, one can choose the model that has the smallest number of non-zero coefficients among those that have out-of-sample error within one-standard deviation of the minimum (Hastie, Tibshirani, and Friedman, 2009, Chapter 7). This is known as the *one standard deviation rule*. The choice of the number k is usually made taking into account the bias-variance trade-off (James et al., 2013, Chapter 5). The parameter α can be tuned using CV or chosen so as to give the desired number of selected variables (Zou and Hastie, 2005).

2.1.2 *Penalised likelihood models for multi-omic data*

In the context of the integration of multiple 'omic datasets, prediction can be done for a variety of purposes. For instance, diagnostic models can be trained to detect whether a person has a certain disease or not, while prognostic models can be used to predict the mortality risk for people suffering from a given condition.

The integration of multiple 'omic datasets in the context of prediction, however, cannot be done via the classical methods for penalised logistic regression such as those presented in the previous section, but requires the development of novel statistical methods. Indeed, previous research has shown that applying classical logistic regression with EN penalty to these datasets can lead to poor results (Liu et al., 2018). The main ideas behind the methods available in the literature are illustrated below.

Predictive models for multi-omic data have also been proposed in the Bayesian literature (Wang et al., 2013; Velten and Huber, 2018). However, these models require a specific set of multi-omic layers, which does not correspond the one we have available in our motivating example. Therefore, we do not consider them here.

A two-step approach

One of the first examples of predictive models for multi-omic datasets is that of Zhao et al. (2015), in the context of cancer prognosis. First, they apply LASSO regression to a multi-omic dataset in order to do variable selection in each layer separately, choosing the tuning parameter λ so as to select at least ten variables in each layer. They then use the selected variables in a l_2-penalised Cox regression model. Boulesteix et al. (2017) criticised this method because it does not take into account the correlations between variables belonging to different layers. However, we show later in this chapter (Section 2.3) that, in the context of binary class prediction, two-step approaches can achieve similar performances to more sophisticated ones, while allowing to select a reasonable number of features in each layer.

Integrative LASSO with penalty factors (IPF-LASSO)

Boulesteix et al. (2017) instead developed a bespoke penalised regression method for multi-omic data. It is similar to a LASSO regression, but it assigns a different penalty to each layer. This approach is called *integrative LASSO with penalty factors* (IPF-LASSO). Denoting by M the number of layers and by X_m each layer's data matrix, where $m = 1, \ldots, M$, Boulesteix *et al.*, like us, are interested in those situations where each layer has the same N observations and a different set of P_m features, i.e. $X \in \mathbb{R}^{N \times P_m}$, and the rows in each matrix X_m correspond to the same statistical units. Let $\beta_j^{(m)}$ be the regression coefficient for the

jth feature of the mth layer. IPF-LASSO tries to find the optimal set of coefficients $\boldsymbol{\beta} = [\beta_1^{(1)}, \ldots, \beta_{P_1}^{(1)}, \ldots, \beta_1^{(M)}, \ldots, \beta_{P_M}^{(M)}]$ such that

$$\min_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^{P_1+\cdots+P_M}} -l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}) + \sum_{m=1}^{M} \lambda_m \|\boldsymbol{\beta}^{(m)}\|_1.$$

Boulesteix *et al.* suggest choosing the penalty parameters by a double CV approach. First, for each candidate set of penalties $\lambda_2, \ldots, \lambda_M$, all the predictors are rescaled as follows:

$$x_{ij}^* = \frac{x_{ij}^{(m)}}{\lambda_m / \lambda_1},$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, P_m$ (note that the features in the first layer remain unchanged). Thanks to this, the same penalty λ_1 can be applied to all the scaled variables, and the parameter λ_1 is estimated via CV in the standard way. The candidate set of penalties $\lambda_2, \ldots, \lambda_M$ that gives the best prediction performance is then selected together with the corresponding value of λ_1 found via CV. How to choose the candidate penalty factors $\lambda_2, \ldots, \lambda_M$ remains an open question; the authors pick a grid of predefined values 2^k, $k = -a, -(a-1), \ldots, 0, \ldots, a-1, a$, where a is an integer that varies between 3 and 6 depending on the application. The limitation of this approach is that the computational burden increases very quickly with the number of layers, making it impossible to explore a large set of possibilities for the penalty terms. Boulesteix *et al.* apply this method to a wide range of simulation settings, as well as real datasets on acute myeloid leukæmia (The Cancer Genome Atlas Research Network, 2013) and breast cancer (Hatzis et al., 2011) where the outcomes are overall survival time and relapse-free survival time respectively.

Multi-tuning parameter elastic-net (MTP-EN)

Similarly to what Boulesteix *et al.* did for LASSO, Liu et al. (2018) show that, if the number of informative features is not the same in each dataset, fitting EN regression models with different penalties for each dataset yields better predictions. They do so by defining a multi-tuning parameter elastic-net regression (MTP-EN). For simplicity, let

$$N(\beta) = (1-\alpha)\|\boldsymbol{\beta}\|_2 + \alpha\|\boldsymbol{\beta}\|_1.$$

Then, the regression parameters of MTP-EN are found by solving the penalised regression problem

$$\min_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^{P_1+\cdots+P_M}} -l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}) + \lambda_1 N(\boldsymbol{\beta}_1) + \cdots + \lambda_M N(\boldsymbol{\beta}_M),$$

where the same parameter α is used for each layer. This corresponds to fitting a weighted EN model

$$\min_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^{P_1+\cdots+P_M}} -l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}) + \lambda_1 N_w(\boldsymbol{\beta})$$

where

$$N_w(\boldsymbol{\beta}) = \alpha \sum_{p=1}^{P_1+\cdots+P_M} w_p |\beta_p| + (1-\alpha) \sum_{p=1}^{P_1+\cdots+P_M} w_p \beta_p^2,$$

and the weights w_p are $\mathbf{w} = [1, \ldots, 1, \lambda_2/\lambda_1, \ldots, \lambda_2/\lambda_1, \ldots, \lambda_M/\lambda_1, \ldots, \lambda_M/\lambda_1]$. This model too is shown to work well for diagnostic and prognostic purposes in cancer studies.

Both IPF-LASSO and MTP-EN can be easily fitted using the `glmnet` package (Friedman, Hastie, and Tibshirani, 2010; R Core Team, 2020), specifying the relative weight of each feature in the `penalty.factor` argument of the `cv.glmnet` function. However, there remains the problem of choosing the penalties of all 'omic layers except the first one.

Integrative elastic-net with penalty factors (IPF-EN)

IPF-LASSO has recently been extended by Zhao and Zucknick (2020) to combine it with the tree-guided group LASSO of Kim and Xing (2012) for which the grid search-type approach proposed by Boulesteix et al. (2017) is not a viable option, given its high computational cost. Therefore, they use the *efficient parameter selection via global optimisation* (EPSGO) algorithm of Fröhlich and Zell (2005) instead (details of the algorithm are given in Section 2.1.3). In the same manuscript, Zhao and Zucknick (2020) also give a more flexible formulation of the structured penalised regression model of Liu et al. (2018) that allows different values of the parameter α for each layer:

$$\min_{\beta_0 \in \mathbb{R},\, \boldsymbol{\beta} \in \mathbb{R}^{P_1+\cdots+P_M}} -l(X, \mathbf{y}; \beta_0, \boldsymbol{\beta}) + \lambda_1 N_1(\boldsymbol{\beta}_1) + \cdots + \lambda_M N_M(\boldsymbol{\beta}_M)$$

where $N_m(\boldsymbol{\beta}_M) = (1-\alpha_m)\|\boldsymbol{\beta}_m\|_2 + \alpha_m \|\boldsymbol{\beta}_m\|_1$. Zhao and Zucknick (2020) call the model of Liu et al. (2018) *sIPF-EN* (where the "s" stands for "simple"), and this new, more general one, *IPF-EN*. This is the naming convention that is used in the remainder of this thesis.

2.1.3 *Efficient parameter selection via global optimisation (EPSGO)*

We present here in more detail the EPSGO algorithm mentioned above, since it is widely used in the simulation studies reported in the next section. Before doing so, we give a short introduction to Gaussian processes (GPs), which are not only

central to the EPSGO algorithm, but also useful to introduce some of the concepts that are key to the kernel methods presented in Chapter 3.

Gaussian processes

Following the definition of Rasmussen and Williams (2006),

Definition 2.1 *A* Gaussian process *is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

A GP is fully specified by its mean function $\mu : \mathcal{X} \to \mathbb{R}$ and covariance function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Here we assume that $\mathcal{X} \subseteq \mathbb{R}^D$.

Given a finite sample of points $\mathbf{x} = [x_1, \ldots, x_N]$, the GP provides a probabilistic model for the function $q(\mathbf{x})$. For this reason, Gaussian processes are often used as priors on functions. We indicate this by

$$q \sim \mathcal{GP}(\mu, \kappa).$$

For simplicity, the mean function is often considered to take value zero on the entire domain, without loss of generality. The choice of the covariance function is instead crucial. From Definition 2.1, it is easy to see that, in order to be a valid covariance function, κ must be

- *symmetric*, i.e. such that $\kappa(x, x') = \kappa(x', x)$ for all $x, x' \in \mathcal{X}$ and
- *positive semi-definite* (PSD), i.e. such that for any finite set of points $x_1, \ldots, x_n$ the matrix Σ with ijth entry $\Sigma_{ij} = \kappa(x_i, x_j)$ is PSD.

If κ is a valid covariance function, the matrix Σ is called *covariance matrix*.

There exist many covariance functions that satisfy these requirements (see Rasmussen and Williams, 2006). Here we consider the Gaussian covariance function, defined as

$$\kappa_{\text{Gaussian}}(x, x') = \exp\left\{-\frac{|x - x'|^2}{2l^2}\right\},$$

where l is an hyperparameter defining the *characteristic length scale*. Various methods exist to pick the values of the hyperparameters of the covariance function. For instance, one can use CV, or maximise the marginal likelihood of the data (Rasmussen and Williams, 2006, Chapter 5). In Figure 2.1 are represented the predictive distributions obtained conditioning on the same data, for different parameters of the characteristic length scale.

Tipically one does not observe exactly $q(\mathbf{x})$, but has instead available noisy observations of $q(\mathbf{x})$, which we denote as

$$y = q(\mathbf{x}) + \epsilon,$$

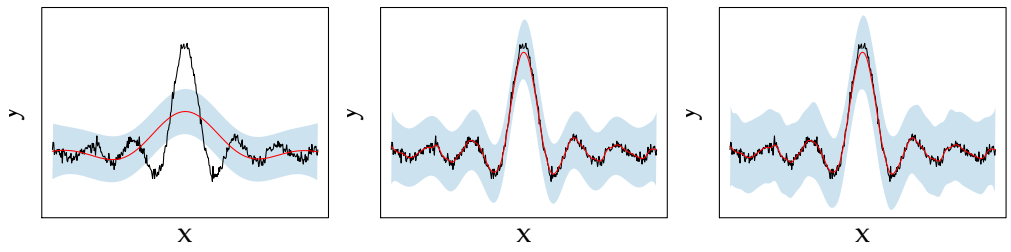


FIGURE 2.1: Predictive mean (in red) ± 2 standard deviation credible intervals (shaded area) of a Gaussian process fitted to a dataset (in black). On the left, the value of characteristic length scale is too low, on the right, too high, and in the middle it is selected via a heuristic implemented in the R package `kernlab`.

where ϵ is independent and identically distributed (i.i.d.) noise with variance σ_ϵ^2. Equivalently, we can write

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \kappa(\mathbf{x}, \mathbf{x}) + \sigma_\epsilon^2 \mathbb{I}_N\right)$$

where $\mathbb{I}_N$ is the $N \times N$ identity matrix.

Using standard properties of Gaussian distributions, one can show that, given a set of N data points $\mathcal{D}_N = \{d_1, \ldots, d_N\}$, where $d_n = (x_n, y_n)$, the predictive distribution conditioned of $\mathcal{D}_N$ is also Gaussian:

$$q(\mathbf{x}^{\text{new}})|\mathbf{x}^{\text{new}}, \mathcal{D}_N \sim \mathcal{N}\left(\mu_N(\mathbf{x}^{\text{new}}), \Sigma_N(\mathbf{x}^{\text{new}}, \mathbf{x}^{\text{new}})\right).$$

for any finite set of points $\mathbf{x}^{\text{new}} = [x_1^{\text{new}}, \ldots, x_N^{\text{new}}]$ where the conditional mean vector μ_N and covariance matrix K_N are available in closed form (Rasmussen and Williams, 2006, Chapter 2). From Definition 2.1, it follows that the posterior distribution q is a Gaussian process:

$$q|\mathcal{D}_N \sim \mathcal{GP}(\mu_N, k_N).$$

Online Gaussian processes

The predictive distribution can also be updated in an iterative fashion using Bayes' rule. Having observed N data points, once a new data point d_{N+1} is available, the updated predictive distribution is:

$$p_{N+1}(y^{\text{new}}|d_{N+1}) = \frac{p(d_{N+1}|y^{\text{new}})\hat{p}(y^{\text{new}})}{\int p(d_{N+1}|y)\hat{p}(y)dy}.$$

Here the integral in the denominator is intractable, so the predictive distribution after observing N points is denoted by $\hat{p}_N$ to indicate that only an approximation

is available. Both the expected model value $\hat{\mu}_N(x)$ and the estimated variance of the model $\hat{\sigma}^2_N(x)$ can be evaluated by recursive formulæ that can be easily updated as soon as a new data point is available (Csató and Opper, 2002).

The EPSGO algorithm

The EPSGO algorithm was initially developed by Fröhlich and Zell (2005) to efficiently tune the parameters of support vector machines (SVMs) and subsequently used by Sill et al. (2014) to select the parameters α and λ of EN (Equation 2.3). The implementation of Sill *et al.* is used in the R package `IPFStructPenalty` of Zhao and Zucknick (2020) to tune the parameters of sIPF-EN and IPF-EN.

The idea is to reframe the task of tuning the model parameters as an optimisation problem. Denote by $\mathcal{A}$ the parameter space of the model of interest and by $q : \mathcal{A} \subseteq \mathbb{R}^D \to \mathbb{R}$ a measure of the quality of the model. In the case of logistic regression models, this can be the out-of-sample MR. The goal is to find the parameters $\mathbf{a}^*$ such that

$$\mathbf{a}^* = \arg\min_{\mathbf{a}\in\mathcal{A}} q(\mathbf{a}).$$

To do so, the EPSGO algorithm models the prior on the error surface q on the parameter space $\mathcal{A}$ as a Gaussian process.

In the first step of the algorithm, some points are sampled from the parameter space that are used to fit an online GP. In order to obtain a good coverage of the parameter space $\mathcal{A}$, the Latin hypercube sampling strategy of McKay, Beckman, and Conover (1979) is used. The recommended number of points to be sampled at this stage is $10D$ (McKay, Beckman, and Conover, 1979).

Once the online GP has been fit on a set of N points, the improvement function is defined as

$$I(\mathbf{x}) = \max\left\{q_{\min} - Y, 0\right\}.$$

where $q_{\min}$ indicates the smallest value of the function q observed up to the current iteration, N, and $Y \sim \mathcal{N}(\hat{\mu}_N, \hat{\sigma}_N)$. A new point $\mathbf{a}_{N+1}$ in the parameter space $\mathcal{A}$ is then chosen so as to maximise the expected improvement criterion of Jones, Schonlau, and Welch (1998):

$$\mathbf{a}_{\text{new}} = \arg\max_{\mathbf{a}\in\mathcal{A}} \mathbb{E}[I(\mathbf{x})].$$

and the online GP is updated after evaluating the error surface at $\mathbf{a}_{N+1}$. This procedure is repeated until convergence is reached (see Algorithm 2.1).

Algorithm 2.1: Efficient parameter selection via global optimisation (EPSGO).

Input : Function $q : \mathcal{A} \to \mathbb{R}$
$\mathbf{l}, \mathbf{u}$ parameter bounds
Initialise: $D = \dim(\mathcal{A})$.

Create $10D$ sample points $\mathbf{a}_1, \ldots, \mathbf{a}_N$ in $[\mathbf{l}, \mathbf{u}]$ using Latin hypercube sampling
Compute $q(\mathbf{a}_i)$, $i = 1, \ldots, N$
$q_{\min} = \min_i q(\mathbf{a}_i)$
$\mathbf{a}^* = \arg\min_i q(\mathbf{a}_i)$
Train online GP
$n_{\text{eval}} = 10D$
repeat
  $\mathbf{a}_{\text{new}} = \arg\max_{\mathbf{a}} \mathbb{E}[I(\mathbf{a})]$
  Compute mean and standard deviation of $\mathbb{E}[I(\mathbf{a})]$
  $q_{\text{new}} = q(\mathbf{a}_{\text{new}})$
  if $q_{new} < q_{min}$ **then**
    $q_{\min} = q_{\text{new}}$
    $\mathbf{a}^* = \mathbf{a}_{\text{new}}$
  end
  Update online GP
  $n_{\text{eval}} = n_{\text{eval}} + 1$
until *convergence*;

Output : $q_{\min}, \mathbf{a}^*, n_{\text{eval}}$

2.2 TWO-STEP APPROACHES

We are interested in predictive models that allow us to simultaneously make predictions and determine which variables are relevant for the problem at hand, while being flexible on the number of selected variables. For this reason, we focus on EN-type methods only. In particular, we propose two ways of doing penalised logistic regression on multi-omic data: (i) separate EN on each layer with fixed α; and (ii) separate EN on each layer where α is selected via EPSGO.

While the EN methods used here are not new, their application to multi-omic data in the way that we propose here is novel. Most importantly, the systematic assessment of the performance of these methods and the comparison to their competitors, which are presented in Section 2.3, has never been performed before.

Step 1: variable selection

- *Separate EN on each layer with fixed α.* First, a variable selection step is performed on each 'omic layer separately using EN with a fixed value of α. The regression models are fitted using the `glmnet` R package and the values of λ in each model are selected via CV. The value that minimises the out-of-sample error is selected. The value of α should be chosen depending on the particular application; here we explore how the performances of the method change for different values of α. For our simulation studies and real data analysis we find $\alpha = 0.1$ to be a reasonable value.
- *Separate EN on each layer, α selected via EPSGO.* The difference between this method and the previous one is that the EPSGO algorithm is used in the first step to pick an optimal value for α in each 'omic layer. This can be convenient when the user does not have a particular preference for the value of α. However, we show in Section 2.5.1 that this approach is not always preferable to the previous one.

Step 2: fitting the predictive model

This step is common to both approaches: the variables selected in the first step are used to build a predictive model using ridge-penalised logistic regression. Again, the `glmnet` R package is used and the value of λ that minimises the CV misclassification rate is picked.

2.3 SIMULATION STUDY

We perform a simulation study in order to compare the two approaches presented in the previous section to their main competitors: *naïve EN* and *sIPF-EN*, detailed below. To this end, we modify the implementations of naïve EN and

sIPF-EN of the R package `IPFStructPenalty`, which currently only handles linear regression, in order to do logistic regression. The two other methods are implemented from scratch, heavily relying on the `glmnet` and `IPFStructPenalty` R packages. We also consider a univariate approach. The code used to produce all the results presented below is available at https://github.com/acabassi/logistic-regression-for-multi-omic-data.

- *Naïve EN.* This is the original EN algorithm (Equation 2.3) applied to the matrix X obtained by concatenating all the 'omic datasets (Figure 1.2). We make use of the EPSGO algorithm to automatically select the best value of α, while λ is chosen via CV.
- *sIPF-EN.* As mentioned in Section 2.1.2, this is a variation of EN that assigns different penalty factors λ to each layer, but selects the same value of α for each of them (Zhao and Zucknick, 2020).
- *Univariate approach.* For each 'omic variable, a logistic regression model is built where the only predictors are the variable of interest, and, where appropriate, any covariates that are known to be related to the outcome and therefore are always included in the model. If the null hypothesis that the regression coefficient of the variable of interest should be zero is rejected, then that variable is selected. A ridge-penalised regression model is then built using all the selected variables as well as the covariates that are always included in the model.

Simulation settings

Our simulation settings are similar to those of Boulesteix et al. (2017). We generate three layers of data for each experiment, with $N = 100$ observations each. The first layer represents a set of clinical covariates that are known to be related to the outcome of interest, and for this reason are not penalised. The other two layers represent two 'omic datasets with varying numbers of covariates and proportions of covariates that are correlated with the outcome. We denote the number of non-penalised covariates by P_N, the number of variables in the first and second penalised layers by P_1 and P_2 respectively. Each has a small number of relevant variables, denoted by P_1^r and P_2^r respectively.

In each dataset, the responses are drawn independently from a Bernoulli distribution with parameter $\tau = 0.5$. The variables are then drawn from the following multivariate Gaussian distributions:

$$[X_1, \ldots, X_{P_N+P_1+P_2}] \,|Y = 0 \sim \mathcal{MN}(\mathbf{0}_{P_N+P_1+P_2}, \Sigma),$$
$$[X_1, \ldots, X_{P_N+P_1+P_2}] \,|Y = 1 \sim \mathcal{MN}(\boldsymbol{\mu}_{P_N+P_1+P_2}, \Sigma),$$

where

$$\boldsymbol{\mu} = [\beta_1, \ldots, \beta_1, 0, \ldots, 0, \beta_2, \ldots, \beta_2, 0, \ldots, 0]$$

with P_1^r elements of $\boldsymbol{\mu}$ equal to β_1 and P_2^r elements equal to β_2. The covariance matrix Σ is either the identity matrix

$$\Sigma_0 = \mathbb{I}_{P_N+P_1+P_2}$$

or a block diagonal matrix similar to the one considered in the simulation studies of Boulesteix et al. (2017) and Zhao and Zucknick (2020) that we indicate with Σ_1. The penalised layers have blocks of correlated variables both within and across layers. All the non-penalised covariates are correlated, but uncorrelated to the penalised ones. That is

$$\Sigma_1 = \left[\begin{array}{c|ccc|ccc} N & & & & & & \\ \hline & A_1 & & & B_{12} & & \\ & & A_1 & & & B_{12} & \\ & & & \ldots & & & \ldots \\ & & & A_1 & & & B_{12} \\ \hline & B_{21} & & & A_2 & & \\ & & B_{21} & & & A_2 & \\ & & & \ldots & & & \ldots \\ & & & B_{21} & & & A_2 \end{array}\right]$$

where $b = 10$, N, A_1 and A_2 are matrices of size $P_N \times P_N$, $P_1/b \times P_1/b$, and $P_2/b \times P_2/b$ respectively with ones on the diagonal and all other elements equal to ρ and B_{12} and B_{21} are matrices of size $P_1/b \times P_2/b$ and $P_2/b \times P_1/b$ respectively with all elements equal to ρ.

We consider the same sets of values for $P_1, P_2, P_1^r, P_2^r, \beta_1, \beta_2$ as Boulesteix *et al.*, reported in Table 2.1. Moreover, we set $P_N = 2$ and $\beta_N = \beta_1$. The value of ρ is equal to 0.4 in all simulation settings, as in Boulesteix *et al.*

	P_N	P_1	P_2	P_1^r	P_2^r	β_N	β_1	β_2
Setting A	2	1000	1000	10	10	0.5	0.5	0.5
Setting B	2	100	1000	3	30	0.5	0.5	0.5
Setting C	2	100	1000	10	10	0.5	0.5	0.5
Setting D	2	100	1000	20	0	0.3	0.3	-
Setting E	2	20	1000	3	10	1	1	0.3
Setting F	2	20	1000	15	3	0.5	0.5	0.5

TABLE 2.1: Values of $P_N, P_1, P_2, P_1^r, P_2^r, \beta_N, \beta_1, \beta_2$ used for the simulation study.

In Appendix A we consider three additional sets of simulation settings. In the first

one, only the two 'omic layers are included in the regression. In the other two, we consider again the same simulation scenarios presented here, but with $P_N = 10$ and 100. We also compare these methods to a different univariate selection method followed by a ridge regression on the selected variables.

Simulation results

Figures 2.2 and 2.3 show the outcome of the simulation studies. For each setting and each regression algorithm, we report the following quantities: the MR on the test set, the MR on the training set, the number of selected variables minus the number of non-penalised covariates, the proportion of selected variables that are among the relevant ones, excluding the non-penalised covariates (precision), the proportion of relevant variables that are selected by the algorithm, excluding the non-penalised covariates (recall).

Figure 2.2 shows that, when the covariates are uncorrelated, all methods have comparable out-of-sample MRs, except in settings E and F where the MR is slightly higher for the naïve approach and slightly lower for sIPF-EN. The within-sample MR is lower for the separate with EPSGO and separate with fixed α methods, suggesting that those two might be prone to overfitting. Concerning the precision, there is no clear pattern throughout settings. On the contrary, the two instances of separate regression on each layer consistently show higher values of the recall. Unsurprisingly, the same two algorithms also select the highest number of variables in all settings. This behaviour is opposite to that of the univariate method, which selects a very low number of variables and therefore has values of the recall always close to zero.

In Figure 2.3, we see that if the covariates are correlated, sIPF-EN has the lowest MR, thanks to the fact that, contrarily to the other methods, it takes into account the correlation between data layers. The only other method that does this is naïve EN, however, assigning the same penalty to all layers puts this method at a disadvantage in the settings where the two layers are highly unbalanced (i.e. settings D, E and F). In those settings, the MR of the two-step approaches is comparable if not better than that of naïve-EN. Again, the within-sample MR suggests that the two algorithms that perform variable selection on each layer separately may be overfitting. As above, the same two methods select the highest number of variables. This is reflected in lower precision and higher recall, on average. As expected, the univariate approach has the worst performance overall.

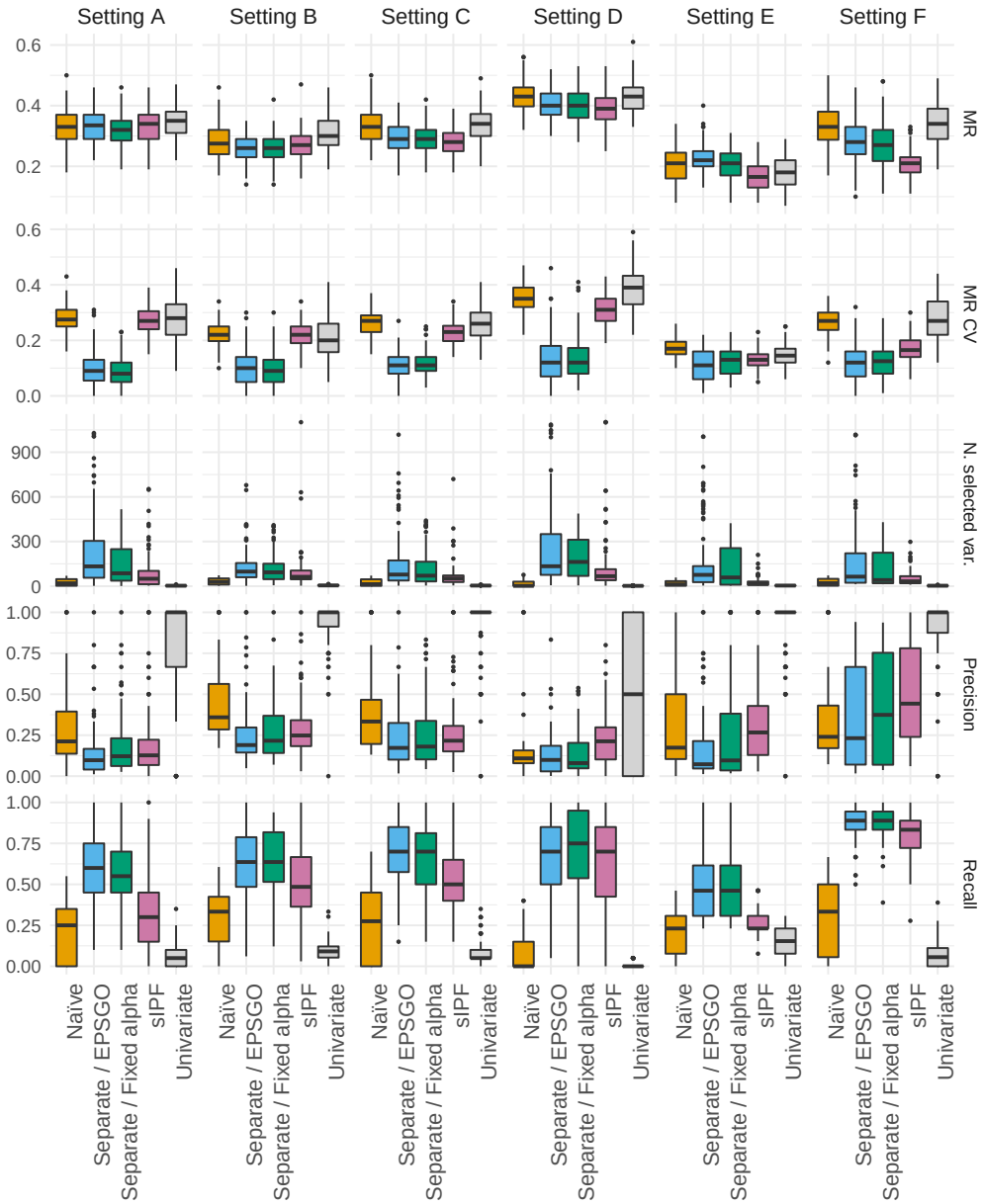


FIGURE 2.2: Simulation study comparing different variants of penalised logistic regression for multi-omic data. The covariance matrix used here is the diagonal matrix Σ_0. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

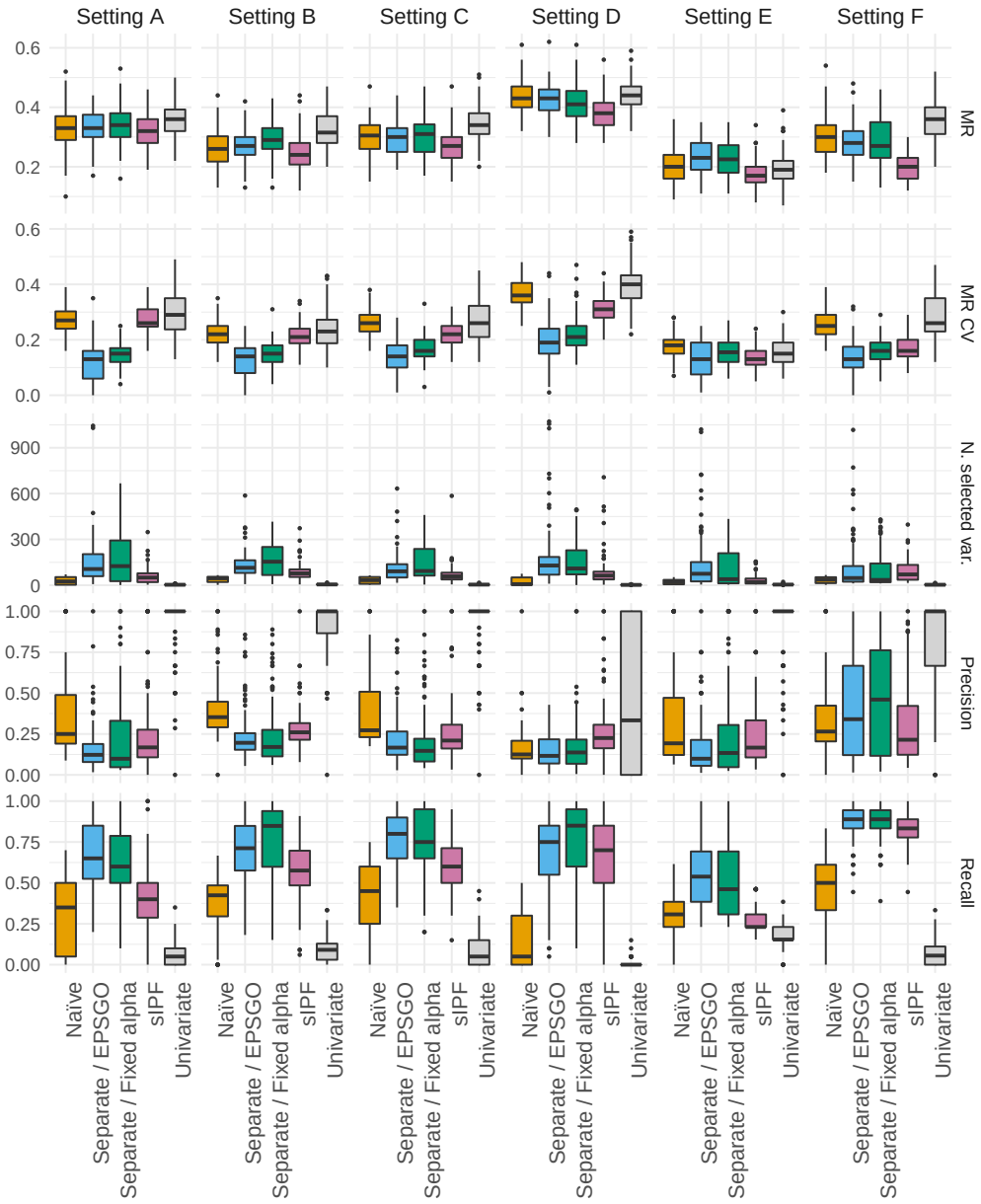


FIGURE 2.3: Simulation study comparing different variants of penalised logistic regression for multi-omic data. The covariance matrix used here is the block matrix Σ_1. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

2.4 CARDIOMETABOLIC SYNDROME DATA

Cardiometabolic syndrome (CMS) is a combination of metabolic dysfunctions such as abdominal obesity, high levels of fasting glucose, elevated blood pressure, low-level inflammation, high level of low-density lipoprotein cholesterol (LDL-C) also known as "bad" cholesterol, and low level of high-density lipoprotein cholesterol (HDL-C) also known as "good" cholesterol (Kirk and Klein, 2009). While the exact causes of CMS are not known, this syndrome has been shown to be associated with higher risk of type 2 diabetes and cardiovascular disease (CVD; see e.g. Grundy et al., 2005). The term CVD encompasses a wide range of conditions, that can be divided into four main categories: coronary heart disease, strokes and transient ischæmic attacks, peripheral arterial disease and aortic disease. According to the World Health Organization, CVD is the leading cause of death and the most common non-communicable disease (*WHO key facts about cardiovascular diseases*).

Here we seek to identify a set of molecular features that characterise CMS by integrating multiple 'omic layers. This can give insights into the molecular mechanisms driving the development of this syndrome and identify relevant biological markers. Moreover, these could be used to stratify the undiagnosed population for their probability of being affected by CMS.

The available data came from three different cohorts:

- *Blood donors.* These are data collected as part of the BLUEPRINT Epigenome[1] work package 10 (WP10) and are related to 184 volunteers recruited amongst the UK's National Health Service Blood and Transplant donors.
- *Patients affected by lipodystrophy syndrome.* Data were collected for ten patients cared for by the National Severe Insulin Resistance Service at Addenbrooke's Hospital in Cambridge. People affected by lipodystrophy syndrome have abnormal lipids distribution and are at high risk of developing CVD. In particular, the individuals considered in this study have familial partial lipodystrophy, which is characterised by loss of subcutaneous fat.
- *Obese individuals.* We have data for ten morbidly obese individuals, i.e. with body mass index (BMI) greater than 40, who were referred for bariatric surgery by the Obesity Clinic of Addenbrooke's Hospital in Cambridge.

Anthropometric, biochemical and 'omic data are available for each individual. Moreover, data collected both before and six months after bariatric surgery were available for obese individuals. Some of the work that was carried out within the framework of the larger study of Seyres et al. (2020) was focused on making use of the data available for the obese individuals before and after surgery to investigate

[1] http://www.blueprint-epigenome.eu/

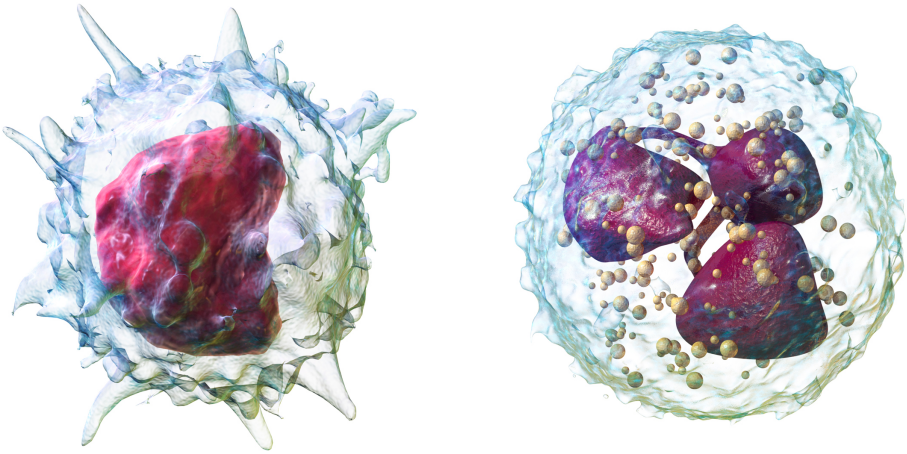


FIGURE 2.4: A monocyte (left) and a neutrophil (right).
Images by Blausen.com staff (2014).

its effects. However, this is beyond the scope of this thesis. Therefore, only the pre-surgery data are considered here.

2.4.1 *'Omic data*

ChIP-seq, RNA-seq data, and DNA methylation data were collected from two types of white blood cells which form an essential part of the innate immune system in humans: monocytes and neutrophils.

Monocytes are white blood cells that can differentiate into macrophages and myeloid lineage dendritic cells. We mentioned above that CVD can be divided into four categories. Three of those, namely coronary heart disease, strokes, and peripheral arterial disease involve atherosclerosis. This makes studying monocytes particularly interesting, since it has been shown that monocytes and monocyte-derived macrophages play an important role in atherosclerosis (van Tuijl et al., 2019).

Neutrophils are white blood cells that act as first responders in case of acute inflammations. It has recently been suggested that neutrophils also play a role in the low-grade inflammation that characterises CMS and that prolonged exposure to this type of inflammation modifies their phenotype (Caielli, Banchereau, and Pascual, 2012; Wright et al., 2010).

In addition to that, metabolites and lipids were measured from plasma. Here, these measurements are of particular interest, since it has been shown that obesity causes a significant perturbation of the plasma metabolome (Cirulli et al., 2019; Moore et al., 2014). Similar conclusions have been made for patients suffering from lipodystrophy (Fiorenza, Chou, and Mantzoros, 2011; Huang-Doran et al.,

2010).

In what follows we give a brief description of how each dataset was collected and preprocessed by our collaborators. Further details can be found in the manuscript of Seyres et al. (2020).

ChIP-seq data

Chromatin immunoprecipitation was performed using the IP-Star Compact Automated System of Diagenode[2] and sequencing was done using Illumina HiSeq 2500[3] or Illumina HiSeq 4000[4]. 67,763 and 49,188 peaks were obtained for monocytes and neutrophils, respectively. A batch effect due to the fact that the sequencing of the donors was performed in a different sequencing centre from the obese and lipodystrophy individuals was corrected using the `Combat` function of the `R` package `sva` (Leek et al., 2019). Details of the batch effect correction method can be found in Johnson, Li, and Rabinovic (2007). After that, low variance peaks and peaks located on sex chromosomes were removed, leaving 25,600 regions in monocytes and 26,300 in neutrophils.

RNA-seq data

For obese and lipodystrophy patients, sequencing was done on Illumina HiSeq 2500 or Illumina HiSeq 4000, whereas RNA-seq data for the WP10 donors were retrieved from the European Genome-phenome Archive of the European Bioinformatics Institute[5]. After preprocessing, 11,370 gene sets were available for monocytes and 24,224 for neutrophils. In the RNA-seq datasets, transcript with more than 25% identical counts were removed. In the monocyte dataset, this resulted in removing 937 out of 11,370 transcripts and in the neutrophil dataset 3,627 out of 24,224. Batch effect correction was performed in the same way as for ChIP-seq data.

DNA methylation data

DNA methylation levels were measured using the microarray-based Infinium Human Methylation 450 BeadChip[6] assays by Illumina, which is a platform for low-cost high-throughput methylation profiling. Methylation is (almost exclusively) found in CpG sites, regions of DNA where a cytosine nucleotide is followed by a guanine nucleotide in the 5′ to 3′ direction (indicated as 5′ – C – p – G – 3′). After preprocessing, the DNA methylation datasets comprised values for

[2] https://www.diagenode.com/en/categories/ip-star

[3] https://emea.illumina.com/systems/sequencing-platforms/hiseq-2500.html

[4] https://emea.illumina.com/systems/sequencing-platforms/hiseq-3000-4000.html

[5] https://www.ebi.ac.uk/ega/home

[6] https://www.illumina.com/documents/products/datasheets/datasheet_humanmethylation450.pdf

26,214 CpG sites for 193 of the available blood samples in monocytes and 21,442 CpG sites for 187 samples in neutrophils.

Metabolite data

Metabolites profiling was performed by *Metabolon Inc.*[7] using their standard protocol. Overall, 988 species were quantified. Each of them was rescaled in order to have a median equal to one. Missing values were imputed using the `impute.knn` function of the R package `impute` (Hastie et al., 2019). Again, batch effect correction was performed in the same way as for ChIP-seq data.

Lipid data

Lipidomics profiling was performed via mass spectrometry using an Advion TriVersa Nanomate[8] interfaced to the Thermo Exactive Orbitrap[9]. The dataset contains measurements of 123 lipids.

2.4.2 *Anthropometric and biochemical parameters*

The anthropometric parameters available for the study are weight, age, and sex. Moreover, plasma biochemistry assays were performed for the leptin, adiponectin, insulin, free fatty acid (FFA), glucose (GLC) and each individual's serum lipid profile, which includes: triglycerides (TG), total cholesterol (TC), HDL-C and LDL-C. In addition, leptin to adiponectin ratio (LAR), homeostasis model assessment of insulin resistance (HOMA-IR) and adipose tissue insuline resistance (ADIPO-IR) were computed for each individual in the study. Detailed explanations of the meaning of these quantities and why these are currently routinely used to assess risk of CMS are given in Appendix A.

Definition of the control group

Based on their clinical parameters, the clinicians involved in this study selected 14 donors as controls. The parameters used to identify the controls are as follows:

- BMI < 25;
- GLC < 5.4 mmol/L;
- TG < 1.7 mmol/L;
- LDL-C < 2.59 mmol/L,
- HDL-C > 1 mmol/L for men and > 1.3 mmol/L for women;
- HOMA-IR score < 2.2.

[7] https://www.metabolon.com/
[8] https://www.advion.com/products/triversa-nanomate/
[9] https://www.thermofisher.com/uk/

These are considered to be healthy people and together are referred to as *controls* in the remainder of this thesis.

2.4.3 *Missing data*

One of the challenges of multi-omic studies is that each data layer often has missing values. There are several reasons why 'omic measurements may not be available, some of which are related to the specific 'omic type at hand. Therefore, each individual will have a different set of observations available. Figure 2.5 shows which individuals had at least one missing value in each 'omic layer. Even though in Seyres et al. (2020) inference is made on all the individuals available, in this chapter we only consider the people who had complete measurements in all layers: 96 out of 205. A different type of analysis is presented in Chapter 5, where all the individuals who have measurements available are included.

In addition to that, not all the anthropometric and biochemical measurements were available for all the people in the study. Table 2.2 shows the number of missing values for each parameter. Considering only the set of people who have no missing 'omic values, the missing values were imputed using the R package `mice` (van Buuren and Groothuis-Oudshoorn, 2011) with default settings. This is a popular package that does multivariate imputation by chained equations (hence the name "mice").

	Blood donors	**Lipodystrophy patients**	**Obese individuals**
# of individuals	184	10	11
Age	0	1	0
Weight	35	0	0
LAR	6	0	0
Glucose	1	0	0
Total cholesterol	2	2	0
TG	1	0	0
HDL-C	1	0	0
LDL-C	7	4	0
ALT	4	0	0
AST	74	0	0
hsCRP	62	0	0
FFA	15	0	0
Insulin	15	0	0
HOMA-IR	16	0	0
ADIPO-IR	36	0	0

TABLE 2.2: Number of missing values for each anthropometric and biochemical parameter before imputation.

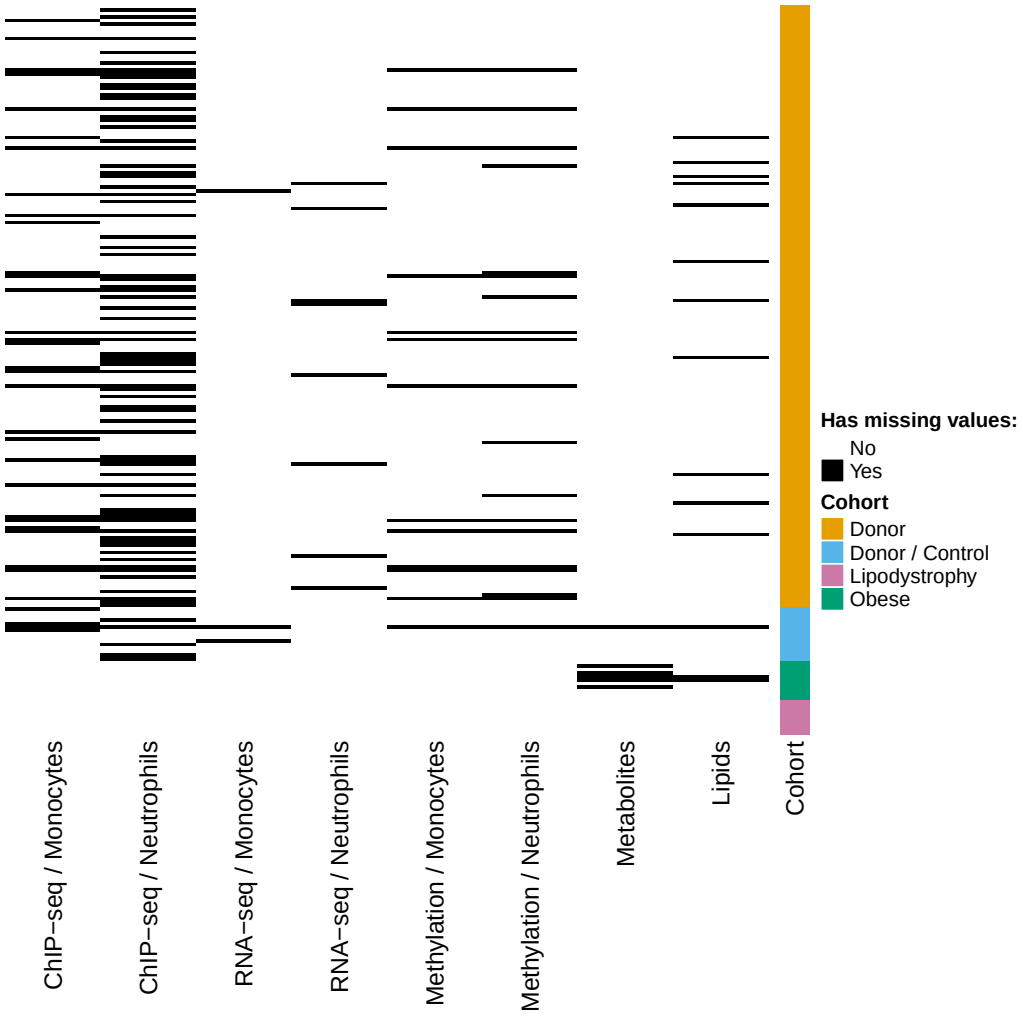


FIGURE 2.5: Missing values in each 'omic layer. Each row corresponds to an individual, each column to a layer. Missing values are indicated in black. The rightmost column shows the three cohorts, where the donors are divided into control donors and others.

2.5 MULTIVARIATE SIGNATURE IDENTIFICATION

This section is dedicated to the multivariate analysis of the CMS data. In Section 2.5.1 we select the variables that help distinguish obese individuals from healthy people, using the penalised logistic regression methods presented in Section 2.2. In Section 2.5.2 we use the selected variables to estimate the probability of each blood donor having CMS.

2.5.1 *Signature identification*

We are interested in identifying combinations of variables that help discriminate between healthy people and CMS patients which, in the remainder of this thesis we refer to as the *molecular signature* of CMS. For this reason, we choose the two algorithms that have the highest recall in our simulation settings, which are those that perform variable selection by training a separate EN model on each layer. Moreover, contrarily to the other methods, these have the advantage of selecting a set of features in each dataset that together are predictive of patient status. This allows us to identify a molecular signature of CMS in each layer.

In this section, first we explain how the available samples are divided into training and test sets. Then, we present the results obtained with a fixed value of α, explaining the rationale behind the choice of the value 0.1. Finally, we comment on the results obtained choosing the value of α as suggested by Zhao and Zucknick (2020) and explain why fixing the value of α turns out to be more convenient for our application.

Training and test sets

We consider the following comparisons:

1. obese individuals versus controls;
2. lipodystrophy patients versus controls.

Each of these defines a different split of our training set into healthy people (*controls*, labelled by "0" in our logistic regression) and patients affected by lipodystrophy or obesity (*cases*, label "1") and helps extracting the most relevant features for each comparison. In this chapter we only present the results obtained for comparison 1, results for comparison 2 are reported in Appendix A.

Separate EN on each layer with fixed α

We use separate EN on each layer with fixed α to identify putative multivariate signatures that discriminate between the considered groups. Before doing so, we

centre and scale each dataset so that all variables have mean 0 and variance 1 across the individuals in which they were measured.

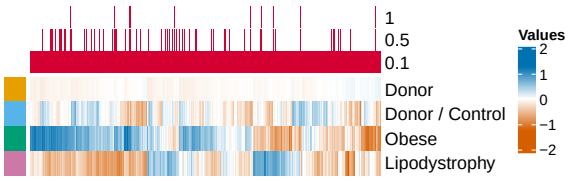
The training set for comparison 1 is formed by the donors who have been selected as controls and the obese individuals. We use 10-fold CV as suggested by Zou and Hastie (2005). To do so, we use the `cv.glmnet` function of the R package `glmnet`. An explanation of why 10 is a good default value for the number of folds to be used in CV can be found in James et al. (2013, Section 5.1.4). Since different CV splits result in different subsets of selected variables, we repeat the CV procedure 1000 times for each layer, and then consider the largest set of selected variables that is selected across the 1000 repeated runs. In Appendix A we compare to a strategy in which we instead choose the set of variables that is selected most often across all the runs. We repeat the analysis with $\alpha = 1$, 0.5 and 0.1. The two highest values of α lead to selecting very few, if any, variables in most layers (Figure 2.6). For this reason, we decide to pick $\alpha = 0.1$. Note that, in this setting, despite giving relative weight of only 10% to the LASSO penalty, a tiny percentage of the available variables is selected.

Figure 2.6 shows, for each 'omic layer, the average value of each selected variable for each category of people: donors who have been selected as controls, the remaining donors, obese individuals and lipodystrophy patients. As we might expect, the average values taken by each variable have opposite sign in the two sets of people used in the training set. Perhaps more interestingly, we note that, while the other donors have average values that are close to zero, lipodystrophy patients take extreme values on most of the selected variables. On top of those values, red bars indicate which variables are selected for each value of α. The variables that are not selected for any value of α are not shown.

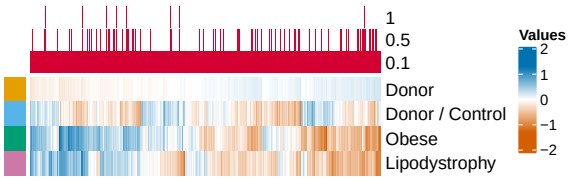
Comparison with separate EN with α selected by EPSGO

We now apply the same strategy as above, except that we let the EPSGO algorithm choose α so as to minimise the MR. The first step of the EPSGO algorithm fails on the lipid data. For this reason, the results presented here only comprise the remaining seven 'omic layers.

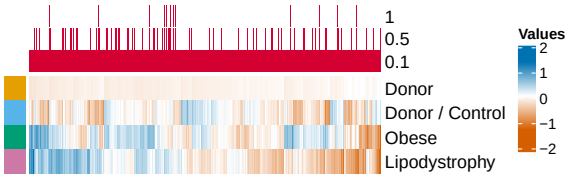
Table 2.3 shows the number of selected variables in each layer by each method, as well as the selected value of α and the number of selected variables that are in common between the two algorithms. The optimal values of α selected via error surface optimisation are all higher than 0.1, except for the metabolite data. Consequently, fewer variables are selected compared to fixing $\alpha = 0.1$. While in some cases this may be a desirable feature, here it makes it difficult to identify molecular signatures, especially for the ChIP-seq data of monocytes, where only one variable is selected.



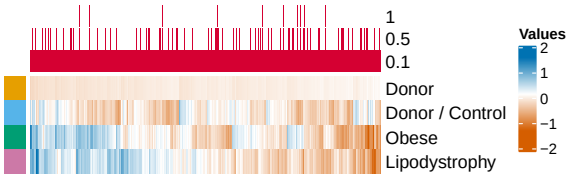
(A) ChIP-seq, monocytes.



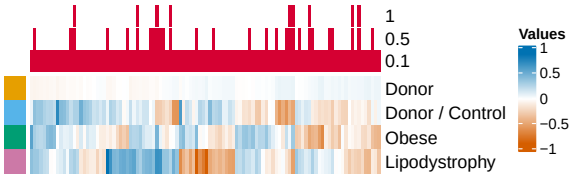
(B) ChIP-seq, neutrophils.



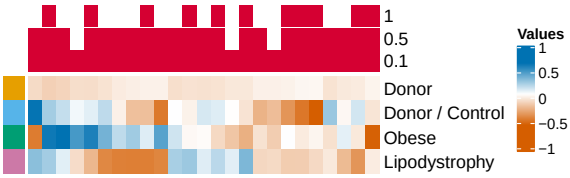
(C) RNA-seq, monocytes.



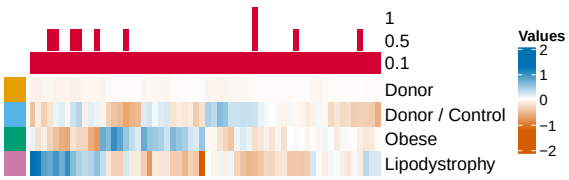
(D) RNA-seq, neutrophils.



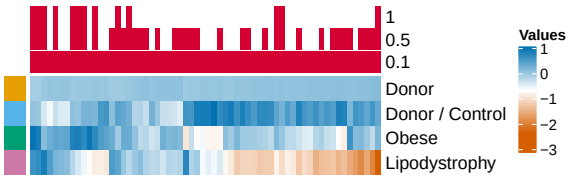
(E) Methylation, monocytes.



(F) Methylation, neutrophils.



(G) Metabolites.



(H) Lipids.

FIGURE 2.6: Variables selected with separate EN and fixed α, for different values of the parameter α. The red bars indicate which variables are selected with $\alpha = 1, 0.5$ and 0.1 respectively. Below are shown the mean values of those variables for the donors and controls and for the lipodystrophy and obese individuals. Only the variables selected for at least one of those values are reported here.

	#var. $\alpha = 0.1$	**#var. EPSGO**	α **EPSGO**	$\cap$
ChIP-seq / Monocytes	428	1	0.59	1
ChIP-seq / Neutrophils	611	40	0.75	29
RNA-seq / Monocytes	425	111	0.96	21
RNA-seq / Neutrophils	592	219	0.13	82
Methylation / Monocytes	106	31	0.75	0
Methylation / Neutrophils	25	54	0.59	5
Metabolites	60	195	0.02	55
Lipids	62	-	-	-

TABLE 2.3: Comparison of separate EN methods. From left to right, are reported: the number of selected variables when $\alpha = 1$, the number of selected variables when α is selected using EPSGO, the value of α picked by EPSGO, and the number of variables that are selected both with fixed α and with α selected via EPSGO.

Therefore, the automatic selection of α via the EPSGO algorithm presents two main disadvantages in applications like ours. The first and most important one is that the algorithm may not always work, as is the case for the lipid data. Secondly, selecting the EN parameter so as to minimise the average out-of-sample error can be quite convenient in some cases, it may not be the best choice in applications where the goal is to select a reasonable number of predictive features in each dataset.

2.5.2 *Probability of being affected by cardiometabolic syndrome*

After performing variable selection on each layer separately, we train a ridge-penalised logistic regression model on the matrix formed by all the variables selected in each layer to compute the probability of each individual to belong to the *case* group. The training and test sets are the same as in Section 2.5.1. Again, we present here only the results obtained for comparison 1 (obese individuals versus controls); the results for comparison 2 (lipodystrophy patients versus controls) can be found in Appendix A.

Figure 2.7 shows, for each person, the probability of belonging to the extreme phenotype group (which in this case comprises the obese individuals). The probabilities estimated on each layer separately are also reported. These are derived by fitting a logistic regression with l_2 penalty on the selected variables only. The ChIP-seq and RNA-seq data give similar predictions, while the lipidomics dataset produces slightly different ones. For the methylation and metabolomics datasets, the probabilities of being a case do not differ greatly among individuals.

It is interesting to note that the lipodystrophy patients have higher probabilities of

belonging to the same class as the obese individuals than the blood donors. This suggests that, on the molecular level, lipodystrophy patients are more similar to obese individuals than the average person. This is not surprising, as those two conditions are characterised by similar biochemical and clinical profiles. On the other hand, some blood donors have very similar predicted values to the obese and lipodystrophy individuals. This may indicate that blood donors can show similar characteristics to those in the extreme phenotype groups, which could provide insights into the pathogenesis of CMS.

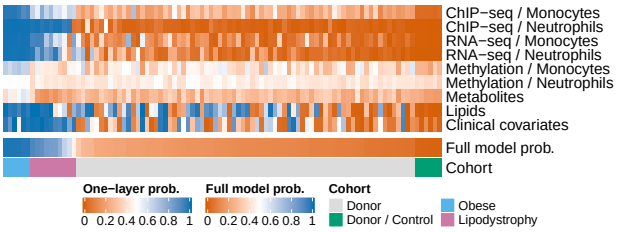
These peculiarities of the results can be better observed by assigning a ranking to each person from 1 to 96 based on their probability of belonging to the class with label "1", where the person with rank 1 has the highest probability. We do this based on the probabilities estimated on each layer separately, and then take the average as the aggregated rank for each person. Note that this combined ranking does not correspond to the ranking implied by the probabilities of class membership obtained using the full ridge-penalised model (Figures 2.7b, 2.7d, and 2.7f). Again, some of the lipodystrophy patients score similarly to the obese individuals. Moreover, we find some donors among the lipodystrophy patients.

Many other ways of combing the rankings could have been considered. The literature on rank aggregation is vast; the first efforts on this topic date back to the XVIII century (de Borda, 1781). This is still a thriving field in modern times, with a wide range of rank aggregation methods being developed for different types of applications, including genomic and multi-omic studies (Blangiardo and Richardson, 2007; Lin and Ding, 2009). An overview is provided by Lin (2010). Due to the fact that, as we have seen, data available for these studies often have missing values, the focus has recently shifted to methods that can handle partial rankings (Aerts et al., 2006; Kolde et al., 2012). A comparative study of such methods has been performed by Li, Wang, and Xiao (2019). However, we find that, in this simple case, taking the average ranks is a sensible choice.

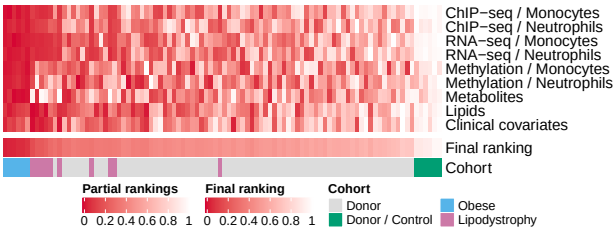
2.6 UNIVARIATE DIFFERENTIAL ANALYSIS

We now compare the multivariate signature identification method described above to a simpler approach that uses univariate tests. For each of the 'omic layers, we use the Mann-Whitney test (Mann and Whitney, 1947) to test for differences in distribution for each 'omic variable.

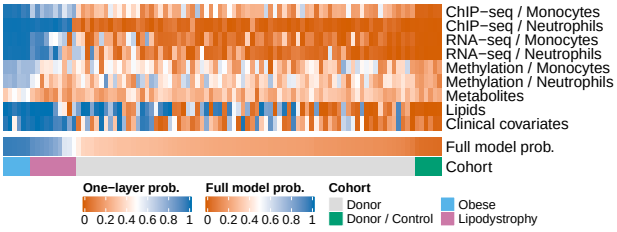
The Mann-Whitney test is a non-parametric test for the difference in the means of two samples. It is used when the underlying distributions are not Gaussian. Let $x_1, \ldots, x_{N_1}$ and $y_1, \ldots, y_{N_2}$ be two independent samples from populations X_1 and X_2 with means μ_1 and μ_2 respectively. X_1 and X_2 are assumed to be continuous distributions that differ only (and at most) in their locations. The hypotheses that



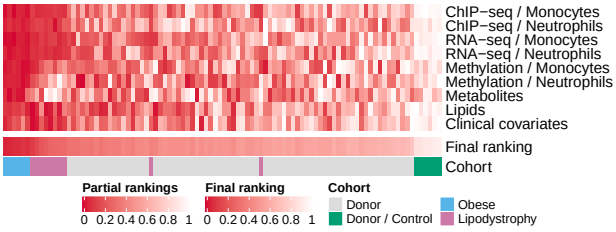
(A) Probability of being a *case*, $\alpha = 0.1$.



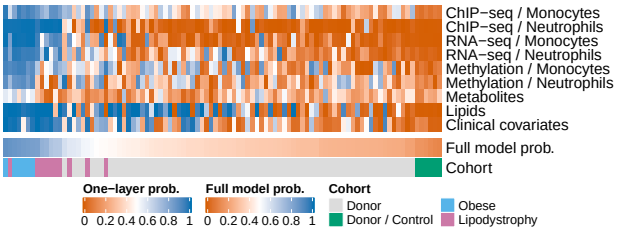
(B) Combined rankings, $\alpha = 0.1$.



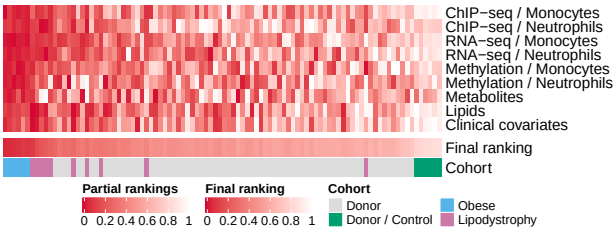
(C) Probability of being a *case*, $\alpha = 0.5$.



(D) Combined rankings, $\alpha = 0.5$.



(E) Probability of being a *case*, $\alpha = 1$.



(F) Combined rankings, $\alpha = 1$.

FIGURE 2.7: Probabilities of belonging to the class of obese individuals and ranking of each person according to those probabilities. Both quantities are shown on each dataset separately and considering all the data types jointly. The model is trained on the obese individuals and control donors. Each column corresponds to one of the individuals who have no missing data, each row corresponds to one of the layers. The columns are sorted by probability of being a case in (A), (C), and (E) and final ranking in (B), (D), and (F). All rankings are divided by the total number of observations.

we want to test are

$$H_0 : \mu_1 = \mu_2 \quad \text{versus} \quad H_1 : \mu_1 \neq \mu_2.$$

To test this, it is sufficient to arrange all the observations in ascending order, assign ranks 1 to $N_1 + N_2$ to them, and, if two or more observations have the same value, use the mean of the ranks that would have been assigned to them if they were not equal. Now, defining W_1 as the sum of the ranks of the observations in the first sample and W_2 the ranks of the observations in the second sample, one would expect that, after adjusting for the cardinality of each set, W_1 and W_2 have similar values if the null hypothesis is true. Therefore, the null hypothesis is rejected if W_1 and W_2 are significantly different. In the case of small samples, the reference distribution is tabulated.

In order to control the *false discovery rate* (FDR), that is the proportion of type I errors in a set of tests, at the 5% level, we adjust the p-values according to the Benjamini-Hochberg procedure. This means that, if $p_{(1)}, \ldots, p_{(K)}$ is our set of p-values in ascending order, for a given α, we find the largest i such that

$$p_{(k)} \leq \frac{i}{k}\alpha,$$

and reject the null hypothesis for all hypotheses having p-values smaller or equal to $p_{(k)}$. This ensures that the expected number of type I errors is less than or equal to α.

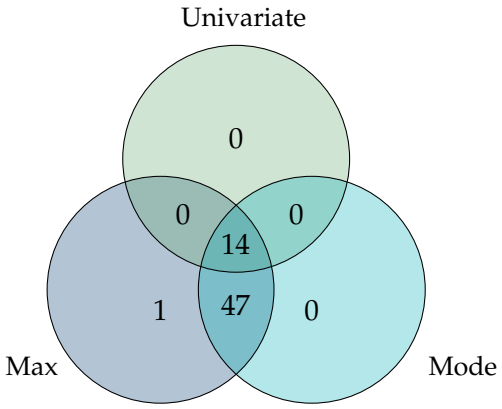
While the multivariate approach selects variables that are jointly useful for prediction and is therefore more permissive, this is not possible with univariate methods. Moreover, this univariate approach is more conservative as it provides control over the FDR.

We repeat the analysis for both comparisons considered above: obese individuals versus controls (comparison 1) and lipodystrophy patients versus controls (comparison 2). The Mann-Whitney test is performed using the `wilcox.test` function of the R package `stats` (R Core Team, 2020); the adjusted p-values are obtained using the `qvalue` function of the Bioconductor[10] package `qvalue` (Storey et al., 2019).

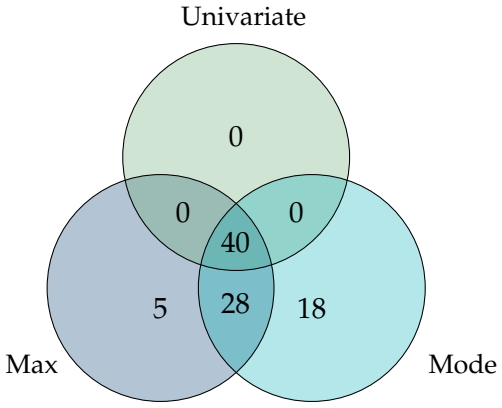
Due to the fact that this test is very conservative and that it cannot take into account synergies between sets of variables that divide the observations into two groups, the number of selected variables is very low with respect to the multivariate setting. For example, only 14 tests are significant in the lipidomics layer, when comparing the obese individuals to the control donors, and no variables show significant differences in any of the other layers. In Figure 2.8 are shown the number of variables that have been selected in the lipidomics layer with the

[10] https://www.bioconductor.org/

multivariate approach (both choosing the largest set of selected variables and the one that is selected most frequently) which are compared to those selected by univariate testing. The corresponding information for the other seven layers can be found in Appendix A.



(A) Obese individuals versus control donors.



(B) Lipodystrophy patients versus control donors.

FIGURE 2.8: Venn diagram showing the intersections between the variables selected in the lipids layer with the multivariate approach with the maximal and modal set of variables and those selected via univariate testing.

2.7 VALIDATION VIA EXTERNAL COHORTS

The last step of the analysis is aimed at validating our findings. Due to the scarcity of multi-omic datasets available for CMS studies, it is not currently possible to try to replicate the entirety of our results with external data. However, we make use of two external cohorts, both containing lipidomics data only, to partially validate the identified CMS signature. In what follows we give a brief description of both cohorts, and the results of the validation analysis.

The Fenland study

The first dataset that we use to validate our findings was collected as part of the Fenland study (Lindsay et al., 2019). We consider a subset of 1,507 participants out of the 12,345 Fenland study participants. These are volunteers without diabetes born between 1950 and 1975, recruited over the Cambridgeshire region between 2005 and 2015. For each person, as well as the lipidomics data, a large number of anthropometric and biochemical parameters are available. We refer the reader to the original manuscript of Lindsay *et al.* for details on each of those.

Among the lipids measured for this study, we choose the eight top lipids that are selected both by our multivariate and univariate analyses. Furthermore, we choose five lipids at random which are not selected by any of our methods to be our control group. We look at the associations between the clinical parameters and the measurements of those lipids that we observe in our data (Figure 2.9). For each pair of lipid and anthropometric or biochemical parameter, this is computed as the regression parameter β_{lipid} of the linear model

$$\text{parameter}_i = \beta_0 + \beta_{\text{age}} \times \text{age}_i + \beta_{\text{female}} \times \mathbb{1}(\text{female}_i) + \beta_{\text{lipid}} \times \text{lipid}_i + \epsilon_i$$

where i indicates the ith person and $\mathbb{1}$ is the indicator function. The significance of the association is tested as

$$H_0 : \beta_{\text{lipid}} = 0 \quad \text{versus} \quad H_1 : \beta_{\text{lipid}} \neq 0$$

using a t-test with test statistic

$$t = \frac{\hat{\beta}_{\text{lipid}}}{\text{SE}(\hat{\beta}_{\text{lipid}})}$$

where SE is the standard error (for further details on the t-tests for regression coefficients see James et al., 2013, Chapter 2). Since one such test is performed for each pair of lipids and clinical parameters, multiple testing correction is required. We control the *family-wise error rate* (FWER), that is the probability of making one or more type I errors in a set of tests, using Bonferroni's correction (Bonferroni,

1936). This amounts to dividing all p-values by the total number of tests. Despite the fact that this correction is extremely conservative, many of the test are significant at level 0.01. The same is not true for the lipids chosen at random.

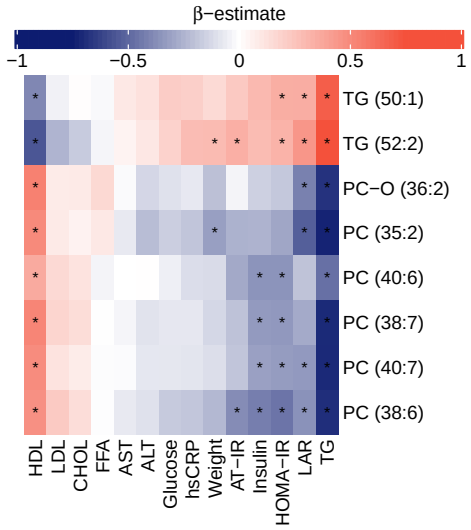
The procedure is repeated for the same two sets of lipids in the Fenland study. We do not have access to this dataset, so the analysis presented here was carried out by Dr Maik Pietzner at the Medical Research Council Epidemiology Unit of the University of Cambridge. In Figure 2.10 we can observe the same patterns as in the previous case: many of the tests are significant and the direction of association between the selected lipids and the clinical covariates is the same. Moreover, here too the strength of association between the lipids selected at random and the clinical covariates is low.

The NASH cohort

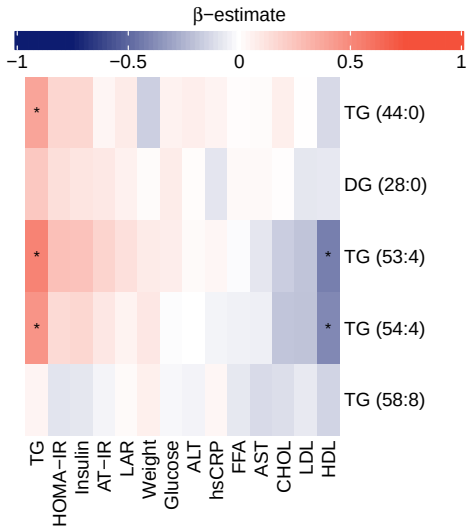
The second dataset used for external validation contains measurements for a cohort of biopsy proven *non-alcoholic steatohepatitis* (NASH) patients (Sanders et al., 2018). NASH is a medical condition that is manifested as an inflammation and damage of the liver caused by a buildup of fat. It can lead to scarring of the liver, which is a life-threatening condition commonly known as cirrhosis. This dataset is formed by 42 patients for whom data are available regarding their BMI, age, sex, glucose, insulin, TG, total cholesterol, LDL-C, HDL-C, AST, ALT, and ALP levels, HOMA2-IR (a new, improved measure of HOMA-IR) index, as well as lipid measurements.

We repeat the same procedure used for the CMS and Fenland data. The output is shown in Figure 2.11. Again, we observe similar correlations here compared to our data. However, in this case the uncertainty on the estimates is very high, due to the fact that the sample size is quite small.

Finally, it is interesting to note that two of the lipids selected by our analysis, PC(38:6) and PC(36:2), have been previously identified in obesity studies (Hall et al., 2017), and that TG(50:1) and TG(52:2) have been linked to non-alcoholic fatty liver disease (Dai et al., 2019) and NASH (Sanders et al., 2018).

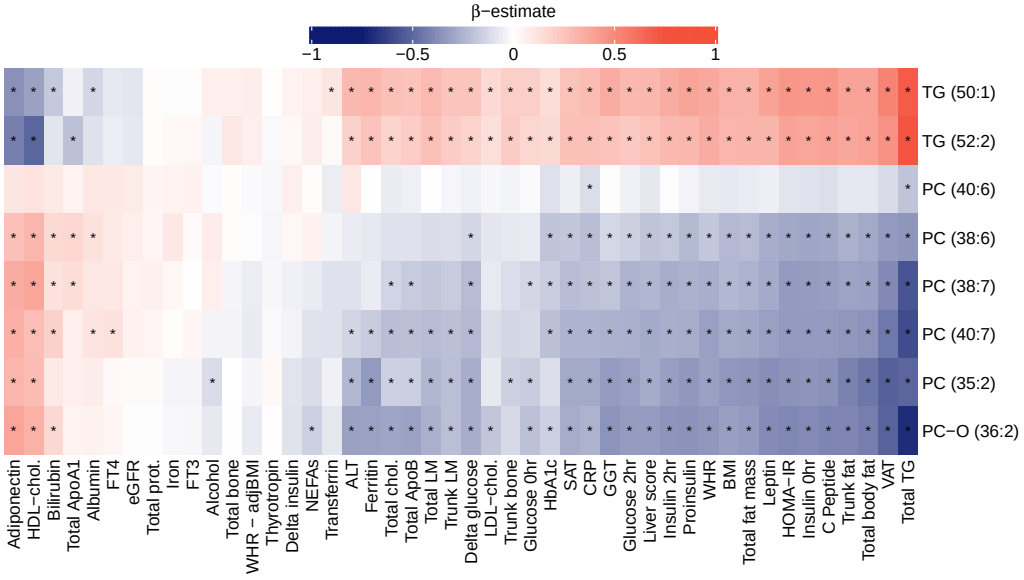


(A) Selected lipids.

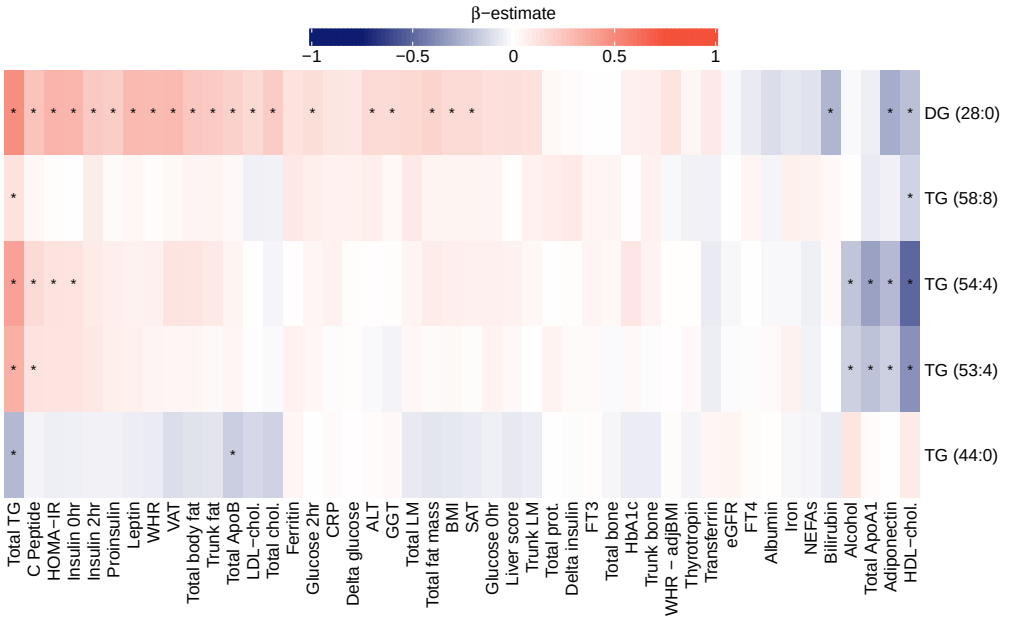


(B) Randomly selected lipids.

FIGURE 2.9: Association between lipid measurements and outcome parameters after correcting for age and sex. In the first plot, the considered lipids are the top eight lipids selected both by our multivariate and univariate analysis, that are also available in the Fenland cohort. In the second plot are shown five lipids that have not been selected in our analysis of the CMS data, chosen at random. Cells marked by an asterisk indicate associations that are significant at level 0.01. Each row corresponds to a lipid and each column to an outcome parameter. To aid visualisation, hierarchical clustering was applied to the rows and columns of the two heatmaps independently, resulting in different column orderings.



(A) Selected lipids.



(B) Randomly selected lipids.

FIGURE 2.10: Association between lipid measurements and outcome parameters after correcting for age and sex. In the first plot, the considered lipids are the top 8 lipids selected both in the univariate and multivariate analysis, that are also available in this cohort. In the second plot are shown five lipids that have not been selected in our analysis of the CMS data, chosen at random. Cells marked by an asterisk indicate associations that are significant at level 0.01. Each row corresponds to a lipid and each column to an outcome parameter. To aid visualisation, hierarchical clustering was applied to the rows and columns of the two heatmaps independently, resulting in different column orderings.

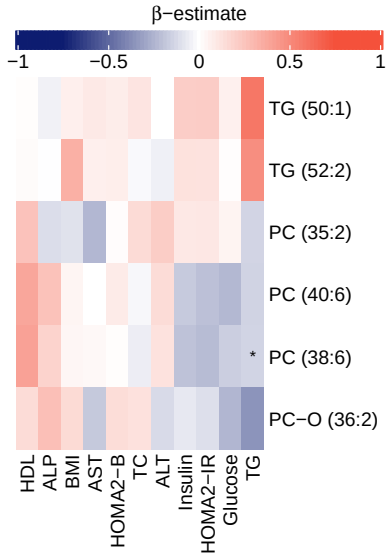


FIGURE 2.11: Association between lipid measurements and outcome parameters after correcting for age and sex in the NASH cohort. The lipids presented here are the same lipids considered in the validation with the Fenland data. Cells marked by an asterisk indicate associations that are significant at level 0.01. Each row corresponds to a lipid and each column to an outcome parameter.

2.8 DISCUSSION

We summarise here the main findings of this chapter, as well as the challenges encountered. Some of the latter provide motivation for the work presented in the following chapters.

2.8.1 *Main findings*

We have proposed two ways of building binary predictive models for multi-omic datasets with the aim of retrieving as many relevant predictive variables as possible. The variables are selected via EN on each 'omic layer separately, allowing the user to either have the freedom to choose the number of selected variable, manually tuning the value of the parameter α, or automatically selecting the value of α that minimises the MR. We have compared these two methods to the two main competitor methods for multi-omic logistic regression that use an EN-type penalty, and to a univariate approach. From our simulation studies, we have concluded that there is no one-size-fits-all approach that is able to achieve low MR in all settings as well as high precision and recall. However, our two suggested methods give higher values of the recall in all simulation settings, so they should be preferred in contexts where the interpretability of the model is key.

We have used the proposed methods to define CMS signatures in eight layers of molecular data, which enable to discriminate between obese individuals and lean, healthy individuals. The disease signatures found in this way have been used to estimate the probability of suffering from CMS for a new set of individuals, including blood donors and lipodystrophy patients. This results in two main findings. First, we noticed that, looking at the selected variables only, lipodystrophy patients have similar molecular profiles to obese individuals in some of the layers. This suggests that different types of CMS may have common molecular pathways. Secondly, some of the blood donors have similar predicted risks (and rankings) to those of the obese and lipodystrophy individuals. This could be due to the fact that those donors are affected by some of the metabolic dysfunctions that characterise CMS. While the dataset considered here is too small to provide results that can be used in the clinic, our study illustrates that, in the presence of a larger set of observations, our model could be used to identify blood donors who show similar molecular profiles to those of CMS patients.

Additionally, we have found that the lipids selected in our analysis are associated with known CMS risk factors, not only in our data, but also in an external cohort. This validates and gives additional credibility to our identified CMS signatures.

Interpretation of the results

While it might be interesting to use models like the one presented here to identify donors affected by CMS, the results presented in this work cannot be used for diagnostic purposes and should be considered as explorative. This is because the sample size is quite small, especially compared to the large number of covariates considered. Widely accepted practices for studies where prediction models are developed, validated or updated for prognostic or diagnostic purposes are those outlined by Moons et al. (2015), which are summarised into the *transparent reporting of a multivariable prediction model for individual prognosis or diagnosis* (TRIPOD) checklist. Moreover, Riley et al. (2020) give precise indications for sample size calculation in clinical prediction models. To build a predictive model that can be used in the clinic, one would need to collect data compliant with those guidelines.

2.8.2 *Challenges*

This work highlights the power of collecting and analysing together multiple 'omic datasets. At the same time, it brings to light some of the main difficulties of working with this type of data.

Small sample size

We mentioned above that the results presented here cannot be used for prognostic purposes. The number of patients and seemingly healthy (control) donors for whom we have data available is very small, especially compared to the large number of covariates available. This is because collecting such a large amount of data for each individual is expensive and time consuming. We reiterate that, despite the encouraging validation results presented in Section 2.7, further studies would be necessary in order to use this model for clinical decision making.

Missing data

As we have seen, data analyses aimed at integrating multiple 'omic datasets are often plagued by large numbers of missing data. In the case of the study presented in this chapter, due to the fact that the considered statistical methods cannot deal with missing data, we are obliged to restrict our analysis to only 96 people out of 205. This represents a great loss of information. To our knowledge, this issue has not yet been addressed for the models presented above and remains an open problem. In Chapter 3, where we present a novel (unsupervised) integrative algorithm for multi-omic data, we make sure that it is able to exploit all the information available for each observation. In Chapter 5 we give a detailed explanation of how missing data can be handled by this method, which is then applied to the CMS data presented here.

Data heterogeneity

Each 'omic layer has different characteristics, which is why treating all layers as one large dataset is not a good idea, as we have seen when applying naïve EN to the CMS data. Here, this problem is addressed either by assigning a different penalty to each layer or by performing variable selection on each layer separately. Depending on the specific application, one may want to choose one or the other approach. This is a key point that we shall take into account when developing novel unsupervised integrative methods for multi-omic data in the next chapters.

Variable selection for multiple high-dimensional datasets

This point is tightly linked to the previous one. The problem of variable selection for large datasets has been thoroughly explored and tackled by many researchers in Statistics. However, when presented with multi-omic datasets, the challenge is to simultaneously perform variable selection on multiple, diverse layers of data. We have seen in Section 2.1.2 that the scientific community is starting to propose new ideas to extend existing supervised variable selection methods to multi-omic studies. We shall see in Chapter 5 how the problem of variable selection for datasets like this one also affects unsupervised algorithms.

Validation and replicability issues

Finally, we have highlighted the challenges of validating the results obtained in multi-omic studies. Thanks to the availability of two external cohorts, we have been able to validate our approach, albeit to a limited degree. A full validation of the results would require an external cohort with more 'omic layers. While this is not currently possible, as the cost of high-throughput techniques decreases, more and more multi-omic datasets are becoming available, making this task easier.

3

MULTIPLE KERNEL LEARNING FOR INTEGRATIVE CLUSTERING OF MULTI-OMIC DATA

Multi-omic integrative clustering is a thriving field, with new statistical and machine learning methods being produced at fast a pace (Rappoport and Shamir, 2018). We have mentioned in the Introduction that these methods seek to identify individuals or genes that have similar characteristics across the 'omic layers. Various successful applications of these approaches to real data have contributed to the popularity of this field of research; most notably, in the context of precision medicine, multi-omic clustering methods have been used to identify novel cancer subtypes (see e.g. Hoadley et al., 2014; Aure et al., 2017; The Cancer Genome Atlas Research Network, 2012).

The motivation for the work presented in this chapter stems from the desire to shed light on the operating characteristics of one such method, *cluster-of-clusters analysis* (COCA). Despite being widely used, especially in the context of tumour subtyping (Hoadley et al., 2014; Aure et al., 2017; The Cancer Genome Atlas Research Network, 2012), the COCA algorithm has never been explicitly laid out and its properties have never been systematically explored. Moreover, its robustness to the inclusion of noisy datasets is unclear.

Here we rigorously benchmark COCA and combine ideas from COCA and *multiple kernel learning* (MKL) in order to propose a new *kernel learning integrative clustering* (KLIC) method that addresses the limitations of COCA. Key to our approach is the result that the consensus matrix returned by consensus clustering (Monti et al., 2003) is a valid kernel matrix. This insight allows us to make use of the full range of multiple kernel learning approaches in order to combine consensus matrices derived from different 'omic datasets.

We perform simulation studies to illustrate our proposed approach and compare it to COCA, as well as other integrative clustering algorithms developed specifically for multi-omic data. Moreover, we show how KLIC and COCA compare in two real data applications. The first one is multiplatform tumour subtyping, where the goal is to find clusters of tumours that have similar molecular characteristics, combining information from different 'omic types. Discovering tumour subtypes can be helpful both to improve prognostic tools and to develop novel

more effective treatment for each subtype. The second application considered in this chapter is transcriptional module discovery. In this case, integrating gene expression and transcription factor binding data, the goal is to find groups of genes that are co-regulated and co-expressed and therefore may have similar biological functions.

Chapter outline

Kernel methods are introduced in Section 3.1, with particular focus on kernel k-means. A review of existing multi-omic integrative clustering algorithms is presented in Section 3.2. The COCA algorithm and its origin are explained in Section 3.3. KLIC is introduced in Section 3.4. In Section 3.5 we assess KLIC on three different types of simulated data, and compare it to COCA and other competitor methods. The two real data applications are presented in Sections 3.6 (multiplatform tumour subtyping) and 3.7 (transcriptional module discovery). Finally, in Section 3.8 we summarise the main findings of the chapter and some of the issues encountered.

3.1 KERNEL METHODS

Much of this chapter concerns kernel methods for clustering of multi-omic data. For this reason, we give here a short introduction to kernel methods.

We start by giving the definition of kernel function (Rasmussen and Williams, 2006, Chapter 4):

Definition 3.1 *Any map δ mapping a pair of inputs $x, x' \in \mathcal{X}$ to $\mathbb{R}$ is a* kernel function.

From this definition it follows that the covariance functions of GPs presented in Section 2.1.3 are nothing but symmetric *positive semi-definite* (PSD) kernel functions. In the context of kernel methods, for instance, the Gaussian covariance function introduced in Section 2.1.3 is usually called a *radial basis function* (RBF) kernel.

Kernel methods proceed by embedding the observations into a higher-dimensional feature space $\mathcal{H}$ endowed with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ and induced norm $\|\cdot\|_{\mathcal{H}}$, making use of a map $\phi : \mathcal{X} \to \mathcal{H}$ (Figure 3.1).

Using this definition, it is possible to prove that any inner product of feature maps gives rise to a symmetric PSD kernel (Shawe-Taylor and Cristianini, 2004, Chapter 3), i.e.

Proposition 3.1 *The map $\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined by $\delta(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$ is a symmetric PSD kernel.*

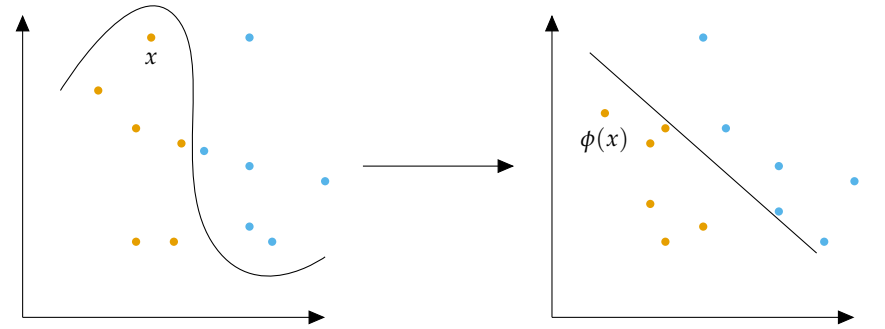


FIGURE 3.1: Using a map ϕ that maps the original data points $x \in \mathcal{X}$ to a different (possibly higher-dimensional) space $\mathcal{H}$, the two classes (orange and blue) become linearly separable. The kernel function δ computes the inner products in the feature space $\mathcal{H}$ directly from the data points in $\mathcal{X}$. Therefore, in many applications the map ϕ does not need to be explicitly known. Figure freely adapted from Shawe-Taylor and Cristianini (2004, Chapter 2).

In this context, given a set of N data points $X = [x_1, \ldots, x_N]$, the $N \times N$ matrix Δ with ijth element equal to the inner product between x_i and x_j in the feature space $\mathcal{H}$ is called *Gram matrix*.

Kernel methods make it possible to model non-linear relationships between data points with a low computational complexity, thanks to the so-called *kernel trick*: in short, many algorithms can be written so that they only require evaluating the kernel function on each pair of data points, without having to explicitly rely on the evaluation of ϕ on each data point (Murphy, 2012, Chapter 14). For this reason, kernel methods have been widely used to extend many traditional algorithms to the non-linear framework, such as *principal component analysis* (PCA; Schölkopf, Smola, and Müller, 1998), linear discriminant analysis (Mika et al., 1999; Roth and Steinhage, 2000; Baudat and Anouar, 2000) and ridge regression (Friedman, Hastie, and Tibshirani, 2001; Shawe-Taylor and Cristianini, 2004).

Moreover, using Mercer's theorem, it can be shown that for any PSD kernel function δ, there exists a corresponding feature map, $\phi : \mathcal{X} \to \mathcal{H}$ (see e.g. Vapnik, 1999). That is,

Theorem 3.1 *For each symmetric PSD kernel δ, there exists a feature map ϕ taking value in some inner product space $\mathcal{X}$ such that $\delta(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{X}}$.*

In practice, it is therefore often sufficient to specify a symmetric PSD matrix Δ representing the similarities between the data points, in order to allow us to apply kernel methods such as those presented in the following sections.

In what follows, we show how the kernel trick can be used to derive the kernel version of the k-means clustering algorithm. This provides an example of

application of the kernel trick, as well as the foundation for the MKL approach presented in Section 3.4. For a more detailed discussion of kernel methods, see e.g. Shawe-Taylor and Cristianini (2004).

Kernel k-means clustering

Before moving on to kernel k-means, we first recall the original k-means clustering algorithm (Steinhaus, 1956). Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ indicate the observed data, with $\boldsymbol{x}_n \in \mathbb{R}^P$ and z_{nk} be the corresponding cluster labels, where $\sum_k z_{nk} = 1$ with $z_{nk} = 1$ if x_n belongs to cluster k, $z_{nk} = 0$ otherwise. We denote by Z the $N \times K$ matrix with ijth element equal to z_{ij}. The goal of the k-means algorithm is to minimise the sum of all squared distances between the data points $\boldsymbol{x}_n$ and the corresponding cluster centroid $\boldsymbol{m}_k$. The optimisation problem is

$$\begin{aligned} \underset{Z}{\text{minimise}} \quad & \sum_n \sum_k z_{nk} \|\boldsymbol{x}_n - \boldsymbol{m}_k\|_2^2 \\ \text{subject to} \quad & \sum_k z_{nk} = 1, \ \forall n, \\ & N_k = \sum_n z_{nk}, \ \forall k, \\ & \boldsymbol{m}_k = \frac{1}{N_k} \sum_n z_{nk} \boldsymbol{x}_n, \ \forall k. \end{aligned} \tag{3.1a}$$

Now we can show how the kernel trick works in the case of the k-means clustering algorithm (Girolami, 2002). Redefining the objective function of Equation (3.1a) based on the distances between observations and cluster centres in the feature space $\mathcal{H}$, the optimisation problem becomes:

$$\begin{aligned} \underset{Z}{\text{minimise}} \quad & \sum_n \sum_k z_{nk} \|\phi(\boldsymbol{x}_n) - \tilde{\boldsymbol{m}}_k\|_{\mathcal{H}}^2 \\ \text{subject to} \quad & \sum_k z_{nk} = 1, \ \forall n, \\ & N_k = \sum_n z_{nk}, \ \forall k, \\ & \tilde{\boldsymbol{m}}_k = \frac{1}{N_k} \sum_n z_{nk} \phi(\boldsymbol{x}_n), \ \forall k. \end{aligned} \tag{3.2a}$$

where we indicated by $\tilde{\boldsymbol{m}}_k$ the cluster centroids in the feature space $\mathcal{H}$. Using the kernel corresponding to ϕ, each term of the sum in Equation (3.2a) can be written

as a function of $\delta(\boldsymbol{x}_i, \boldsymbol{x}_j)$:

$$\begin{aligned}\|\phi(\boldsymbol{x}_n) - \tilde{\boldsymbol{m}}_k\|^2_{\mathcal{H}} &= \langle \phi(\boldsymbol{x}_n) - \tilde{\boldsymbol{m}}_k, \phi(\boldsymbol{x}_n) - \tilde{\boldsymbol{m}}_k \rangle_{\mathcal{H}} \\ &= \langle \phi(\boldsymbol{x}_n), \phi(\boldsymbol{x}_n) \rangle_{\mathcal{H}} - \frac{2}{N_k} \sum_{i=1}^{N} z_{ik} \langle \phi(\boldsymbol{x}_n), \phi(\boldsymbol{x}_i) \rangle_{\mathcal{H}} \\ &\quad + \frac{1}{N_k^2} \sum_{i=1}^{N} \sum_{j=1}^{N} z_{ik} z_{jk} \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}} \\ &= \delta(\boldsymbol{x}_n, \boldsymbol{x}_n) - \frac{2}{N_k} \sum_{i=1}^{N} z_{ik} \delta(\boldsymbol{x}_n, \boldsymbol{x}_i) + \frac{1}{N_k^2} \sum_{i=1}^{N} \sum_{j=1}^{N} z_{ik} z_{jk} \delta(\boldsymbol{x}_i, \boldsymbol{x}_j).\end{aligned}$$

Therefore, there is no need to evaluate the map ϕ at every point $\boldsymbol{x}_i$ to compute the objective function of Equation (3.2a). Instead, one just needs to know the values of the kernel evaluated at each pair of data points $\delta(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j = 1, \ldots, N$.

Defining L as the $K \times K$ diagonal matrix with kth diagonal element equal to N_k^{-1} and Δ the $N \times N$ matrix with ijth entry equal to $\delta(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the optimisation problem (3.2) can be rewritten as a trace maximisation problem (Gönen and Margolin, 2014):

$$\begin{aligned} \underset{Z}{\text{maximise}} \quad & \operatorname{tr}(L^{\frac{1}{2}} Z' \Delta Z L^{\frac{1}{2}}) \\ \text{subject to} \quad & Z 1_k = 1_n, \quad \forall n, k, \\ & z_{nk} \in \{0, 1\}, \quad \forall n, k. \end{aligned} \tag{3.3a}$$

The integrality constraints make this problem difficult to solve. However, the corresponding linear problem obtained by relaxing the integer constraints of Equation (3.3a) to $0 \leq z_{nk} \leq 1$ for all n, k can be solved by performing kernel PCA on the kernel matrix Δ and setting the matrix $H = ZL^{\frac{1}{2}}$ to the K eigenvectors that correspond to K largest eigenvalues (Schölkopf, Smola, and Müller, 1998). The clustering solution can be found by first normalising all rows of H to be on the unit sphere and then performing k-means clustering on the normalised matrix. Other possible approaches to derive a final clustering from H are listed in Shawe-Taylor and Cristianini (2004, Chapter 8).

3.2 INTEGRATIVE CLUSTERING OF MULTI-OMIC DATA

Many existing statistical and computational tools have been applied to the problem of clustering multi-omic data and many others have been developed specifically for this. A recent comprehensive review of integrative clustering algorithms is the one by Rappoport and Shamir (2018) which includes a benchmark of several methods on cancer datasets.

In this section, we review the main integrative clustering algorithms developed

so far specifically for multi-omic data. Table 3.1 contains a summary of existing methods. For clarity of exposition, we classify them into consensus clustering methods, kernel-based algorithms, latent variable models, Bayesian integrative models, network-based methods, and others. We focus particularly on the first two classes of methods, since they are more closely related to the novel algorithm presented here. Alternatively, we have seen in the Introduction that integrative clustering methods can be broadly divided into joint modelling and two-step approaches. The former simultaneously consider all datasets together. The latter, which we consider here, are composed of two steps: first, the clustering structure in each dataset is analysed independently; then an integration step is performed to find a common clustering structure that combines the individual ones.

3.2.1 *Consensus clustering methods*

Methods that belong to this class were initially devised to cluster a single dataset, with the aim of combining multiple partitionings of the observations into one; examples include: the consensus clustering (CC) algorithm of Monti et al. (2003), the ensemble clustering methods of Strehl and Ghosh (2002), and the probabilistic model of consensus of Topchy, Jain, and Punch (2004).

In the field of integrative 'omics for cancer applications, the most popular clustering algorithm is COCA. As we mentioned in the introduction of this chapter, COCA has been widely applied to problems related to tumour subtyping and has grown in popularity since its first introduction by The Cancer Genome Atlas Research Network (2012). COCA proceeds by first clustering each of the datasets separately, and then building a binary matrix that encodes the cluster allocations of each observation in each dataset. This binary matrix is then used as the input to a CC algorithm (Monti et al., 2003; Wilkerson and Hayes, 2010), which returns a single, global clustering structure, together with an assessment of its stability. The idea is that this global clustering structure both combines and summarises the clustering structures of the individual datasets. Despite its widespread use, to the best of our knowledge the COCA algorithm has never previously been systematically explored. In Section 3.3, we elucidate the algorithm underlying COCA, and highlight some of its limitations. We show that one key limitation is that the combination of the clustering structures from each dataset is unweighted, making the output of the algorithm sensitive to the inclusion of poor quality datasets, or datasets that define unrelated clustering structures.

Moreover, Nguyen et al. (2017) developed *perturbation clustering for data integration and disease subtyping* (PINS), a two-step algorithm. In the first step, each dataset X_m, $m = 1, \ldots, M$, is partitioned into K_m clusters. From each clustering, a *connectivity matrix* of size $N \times N$ is defined such that the ijth element is equal to one if samples i and j belong to the same cluster in dataset m, zero otherwise.

The *original similarity matrix* is then defined as the average of all connectivity matrices. If this matrix has more than half of its non-diagonal entries equal to either zero or one, this is interpreted as a strong agreement between the clusterings of all datasets, and therefore the final clustering is performed applying hierarchical clustering on the *strong similarity matrix*, which has ijth element equal to 1 if samples i and j are clustered together in every layer, 0 otherwise. If, however, there is no strong agreement between layers, then a matrix of distances between samples is defined as one minus the original similarity matrix. This matrix of distances is used as the input of a distance-based clustering algorithm to find a global clustering of the samples. The goal of the second step of PINS is to find subclusters that may not have been discovered in the first step. To this end, various indicators are considered, such as the entropy (Cover and Thomas, 2006) and gap statistic (Tibshirani, 2001). Nguyen et al. (2019) developed an improved version of the PINS algorithm, PINSPlus, which is available on CRAN as an R package.

3.2.2 *Kernel-based algorithms*

An alternative class of approaches for integrating multiple 'omic datasets is provided by those based on kernel methods. In these, a kernel function (which defines similarities between different units of observation) is associated with each dataset. These may be straightforwardly combined in order to define an overall similarity between different units of observation, which incorporates similarity information from each dataset. The advantage of using kernels in this setting is that layers of data of different type (e.g. continuous, binary, multinomial, etc.) can be summarised into similarity (i.e. Gram) matrices, which, in contrast to the original data, are directly comparable. Determining an optimal (weighted) combination of kernels is known as *multiple kernel learning* (MKL); see, for example, Lanckriet et al. (2004b), Bach, Lanckriet, and Jordan (2004), Yu et al. (2010), Gönen and Alpaydın (2011), Wang et al. (2017), and Strauß et al. (2020). A challenge associated with these approaches is how best to define the kernel function(s), for which there may be many choices. We shall expand on this topic in Section 3.4.2.

Speicher and Pfeifer (2015) combined the *multiple kernel learning dimensionality reduction* (MKL-DR) approach of Yan et al. (2006) with the *locality preserving projections* (LPP) of He and Niyogi (2004), as well as additional regularisation constraints, to create a new algorithm named rMKL-LPP. In brief, first they solve the

optimisation problem

$$\begin{aligned} \underset{\boldsymbol{\zeta}, \boldsymbol{\psi}}{\text{minimise}} \quad & \sum_{i,j=1}^{N} \|\boldsymbol{\zeta}^T \tilde{\Delta}_i \boldsymbol{\psi} - \boldsymbol{\zeta}^T \tilde{\Delta}_j \boldsymbol{\psi}\|^2 w_{ij} \\ \text{subject to} \quad & \sum_{i,j=1}^{N} \|\boldsymbol{\zeta}^T \tilde{\Delta}_i \boldsymbol{\psi}\| d_{ij} = \text{const.}, \\ & \|\boldsymbol{\psi}\|_1 = 1, \\ & \psi_m \geq 0, \ m = 1, 2, \ldots, M', \end{aligned}$$

where $M' \geq M$, $\tilde{\Delta}_i$, $i = 1, \ldots, N$, are $N \times M'$ matrices where the element in position (j, m) is equal to $\delta_m(\mathbf{x}_i, \mathbf{x}_j)$, w_{ij} is equal to one if i is in the neighbourhood of j and/or vice versa, d_{ij} is equal to $\sum_{n=1}^{N} w_{in}$ if $i = j$, zero otherwise, $\boldsymbol{\zeta} = [\zeta_1, \ldots, \zeta_N]$, and $\boldsymbol{\psi} = [\psi_1, \ldots, \psi_M]$. rMKL-LPP can perform the integration of a large number of kernels, while the l_1 constraint ensures that overfitting is avoided. Thanks to this, they get around the problem of choosing the kernel parameters by considering a set of kernels built with a range of plausible parameter values for each layer. The MKL approach then up-weights the matrices with "high information content" and down-weights the others. The regularisation term is included to avoid assigning non-zero weight to too many kernel matrices. When applying rMKL-LPP to real data, they use RBF kernels with five different parameters for each data type. Their software is available as a web server named web-rMKL (Röder et al., 2019).

Ramazzotti et al. (2018) introduced *cancer integration via multikernel learning* (CIMLR), an extension to multi-omic data of the *single-cell interpretation via multi-kernel learning* (SIMLR) method presented by Wang et al. (2017) and Wang et al. (2018). Given a number of clusters K, CIMLR builds a number of different RBF kernels for each data type, and then combines them to find a weighted similarity matrix $\bar{\Delta}$ that presents a structure that is as close as possible to a block structure with K blocks. This is done by solving the following optimisation problem

$$\begin{aligned} \underset{\bar{\Delta}, R, \boldsymbol{\theta}}{\text{minimise}} \quad & -\sum_{i,j,l} \theta_l \Delta_{ij}^{(l)} \bar{\Delta}_{ij} + \zeta \|\bar{\Delta}\|_F^2 + \gamma \text{tr}\left[R^T (\mathbb{I} - \bar{\Delta}) R\right] + \upsilon \sum_l \theta_l \log \theta_l \\ \text{subject to} \quad & R^T R = \mathbb{I}, \\ & \sum_l \theta_l = 1, \quad \theta_l \geq 0, \\ & \sum_j \bar{\Delta}_{ij} = 1, \quad \bar{\Delta}_{ij} \geq 0, \end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_L]$, ζ and γ are non-negative tuning parameters, and R is an auxiliary low-dimensional matrix ensuring that $\bar{\Delta}$ is of low rank. The suggested number of kernels per layer, which Ramazzotti *et al.* derived empirically, is 55. In particular, they use RBF kernels per data type

in their real data application. The weighted kernel matrix $\bar{\Delta}$ is then used as input for kernel k-means clustering.

In a similar spirit, Mariette and Villa-Vialaneix (2018) suggest three unsupervised multiple kernel learning (UMKL) approaches, sparse-UMKL, full-UMKL, and STATIS-UMKL (named after the STATIS method of Lavit et al., 1994), which can be used to combine multiple kernels into one in a meaningful way. Each method summarises the kernels in slightly different ways. The combined kernel is then used as the input of kernel PCA. Again, the kernels used in the real data application are Gaussian kernels.

Finally, Rappoport and Shamir (2019) developed an approach called *neighbourhood-based multi-omics clustering* (NEMO). Denoting by ν_{im} the k nearest neighbours of $\mathbf{x}_{im}$ within layer m, they define a similarity matrix for each layer $m = 1, \ldots, M$. These similarity matrices are based on the RBF kernel and have element (i, j) equal to

$$\mathrm{S}_m(i,j) = \frac{1}{\sqrt{2\pi}\upsilon_{ijm}} \exp\left\{ -\frac{\|\mathbf{x}_{mi} - \mathbf{x}_{mj}\|^2}{2\upsilon_{ijm}^2} \right\},$$

where υ_{ijm} is a normalising constant defined as

$$\upsilon_{ijm}^2 = \frac{1}{3}\left(\frac{1}{k} \sum_{r \in \nu_{mi}} \|\mathbf{x}_{mi} - \mathbf{x}_{mr}\|^2 + \frac{1}{k} \sum_{r \in \nu_{mj}} \|\mathbf{x}_{mj} - \mathbf{x}_{mr}\|^2 + \|\mathbf{x}_{mi} - \mathbf{x}_{mj}\|^2 \right).$$

The similarity between observations i and j in layer m relative to i and j's nearest neighbours is then defined as

$$RS_m(i,j) = \frac{S_m(i,j)}{\sum_{r \in \nu_{mi}} S_m(i,r)} \mathbb{I}(j \in \nu_{mi}) + \frac{S_m(i,j)}{\sum_{r \in \nu_{mj}} S_m(r,j)} \mathbb{I}(i \in \nu_{mj}).$$

The final clusters are computed as by performing spectral clustering (Ng, Jordan, and Weiss, 2002) on the average of all relative similarity matrices. The R implementation of this method is available as the GitHub package `nemo`.

What all these approaches have in common is that they use predefined closed form kernels (e.g. RBF) whose parameters need to be selected by the user. How to do this best in unsupervised settings is an open question and, as a result, one often has to feed a large number of kernel matrices to these algorithms, which can be quite inefficient. In the next section, we show how this problem can be avoided by defining more meaningful kernel matrices derived from consensus clustering.

3.2.3 *Latent variable models*

One of the first statistical methods applied to integrative clustering for cancer subtypes was *iCluster* (Shen, Olshen, and Ladanyi, 2009). iCluster finds a partitioning of the tumours into different subtypes by projecting the available datasets onto a common latent space, maximising the correlation between data types. The original formulation assumes that all layers have normally distributed data, and can be visualised as follows:

$$\begin{aligned} X_1 &= \Gamma_1 \Pi + \epsilon_1, \\ X_2 &= \Gamma_2 \Pi + \epsilon_2, \\ &\dots \\ X_M &= \Gamma_M \Pi + \epsilon_M, \end{aligned}$$

where Π is the latent component common to all layers, $\Gamma_1, \dots, \Gamma_M$ are the coefficient matrices, and $\epsilon_1, \dots, \epsilon_M$ represent the Gaussian noise. The first version of iCluster only included a LASSO-type penalty on Π; while a subsequent extension (Shen, Wang, and Mo, 2013) introduced EN and fused LASSO (Tibshirani et al., 2005) penalties. Other extensions of iCluster include a variance weighted penalty term, proportional to the error variance associated with each feature so that features with high variance are more strongly penalised (Shen et al., 2012), feature selection using lasso regression Shen, Wang, and Mo (2013), integration of binary, categorical, count, and continuous data types (*iClusterPlus*; Mo et al., 2013), and a fully Bayesian version of the model (*iClusterBayes*; Mo et al., 2018). Implementations of iCluster and iClusterPlus are available on CRAN and Bioconductor respectively.

Zhang et al. (2012) proposed a similar approach, based on non-negative matrix factorisation. They add non-negativity constraints to the matrices Π and Γ_m, $m = 1, \dots, M$ and solve the optimisation problem

$$\begin{aligned} \underset{\Pi, \Gamma_1, \dots, \Gamma_M}{\text{minimise}} \quad & \sum_{m=1}^{M} \| X_m - \Gamma_m \Pi \|_F^2 \\ \text{subject to} \quad & \Pi \geq 0, \\ & \Gamma_m \geq 0, \ m = 1, \dots, M. \end{aligned}$$

Lastly, Wu et al. (2015) suggested to define the likelihood of each type of data X_m conditional on a hidden parameter Π_m and then jointly maximise the full data likelihood

$$l(\Pi) = \sum_{m=1}^{M} l(X_m | \Pi_m)$$

where Π is a matrix formed by stacking Π_m, $m = 1, \ldots, M$. This method is called *low-rank approximation-based multi-omics data clustering* (LRAcluster).

As we shall see later in this chapter, a drawback of these methods is that they assume that the same clustering structure can be found in every data layer.

3.2.4 *Bayesian integrative models*

The first Bayesian integrative models for 'omic data were developed for the specific datasets available in each application. For instance, Liu et al. (2007) and Savage et al. (2010) developed methods to integrate expression and ChIP-chip data, while Rogers et al. (2008) focused on transcriptomic and proteomic expression data, and Yuan, Savage, and Markowetz (2011) on copy number alterations and gene expression data.

The first Bayesian method for integrative clustering that can be used for any type 'omic layers to be developed was *multiple dataset integration* (MDI; see Kirk et al., 2012; Mason et al., 2016). It is based on Dirichlet-multinomial mixture models in which the allocation of observations to clusters in one dataset influences the allocation of observations in another, while allowing different datasets to have different numbers of clusters.

Similarly, *Bayesian consensus clustering* (BCC) of Lock and Dunson (2013) is based on a Dirichlet mixture model that assigns a different probability model to each dataset. Again, tumour samples belong to different partitions, each given by a different data type, but here they also adhere loosely to an overall clustering.

More recently, Gabašová, Reid, and Wernisch (2017) developed *Clusternomics*, a mixture model over all possible combinations of cluster assignments on the level of individual datasets that allows to model different degrees of dependence between clusters across datasets.

Finally, Afrin et al. (2020) introduced a Bayesian directional multi-omic clustering approach that takes into account the directional dependence between 'omic datasets. They use a hierarchical Dirichlet mixture model using copulas (Nelsen, 2007; Jaworski et al., 2010; Trivedi and Zimmer, 2007) to model directional dependencies between layers, where directionality is based on the central dogma (Crick, 1958; Crick, 1970).

The advantage of Bayesian models is that they allow to jointly model multiple layers in a principled way and are particularly suitable to define models not relying on the assumption that all layers have the same clustering structure. On the other hand, inference for these models is often computationally costly, especially considering that 'omic datasets often have large numbers of features.

3.2.5 *Network-based methods*

Wang et al. (2014) developed *similarity network fusion* (SNF). Initially, a network is built for each data layer where the observations are represented by nodes and pairwise correlations between observations form the edges. The networks are then "fused", using a method based on message-passing theory, until convergence to a final network. Spectral clustering is then performed on the final network to identify the global clusters. A more efficient version of SNF is the *affinity network fusion* (ANF) method proposed by (Ma and Zhang, 2018).

A different approach is suggested by Rappoport, Safra, and Shamir (2020), who developed *multi-omic clustering by non-exhaustive types* (MONET). This method aims to circumvent the problem of differing clustering structures by outputting a set of modules, each characterised by a set of samples and a set of 'omics in which those samples are similar. This is achieved with a two-step algorithm. First, a graph is built for each 'omic layer where the nodes are samples and the edges are similarities between samples in that layer. Then modules are detected by looking for subgraphs common to multiple graphs.

3.2.6 *Other methods*

Kim et al. (2015) proposed an *integrative phenotyping framework* (iPF) for multi-omic data. First, all 'omic layers are combined to create one large matrix containing all measurements. The matrix based on the correlations between features is used as input for multidimensional scaling (Cox and Cox, 2008). In this way, the features are mapped to a bidimensional Euclidean space.

Finally, deep learning approaches have recently started to be applied to cancer applications (Levine et al., 2019). In the context of unsupervised methods, notable examples include the multimodal *deep belief network* (DBN) of Liang et al. (2014), the variational autoencoders (VAEs) for cancer data integration of Simidjievski et al. (2019), and the *pathway based multi-modal sparse autoencoders for clustering of patient-level multi-omics data* (PathMe) of Lemsara, Ouadfel, and Fröhlich (2020).

3.3 CLUSTER-OF-CLUSTERS ANALYSIS

COCA (The Cancer Genome Atlas Research Network, 2012) is an integrative clustering method that was first introduced in a breast cancer study by The Cancer Genome Atlas Research Network (2012) and quickly became a popular tool in cancer studies (see e.g. Hoadley et al., 2014 and Aure et al., 2017). It makes use of CC (Monti et al., 2003), an algorithm that was originally developed to assess the stability of the clusters obtained with any clustering algorithm.

Type	Name	Reference
Consensus/perturbation	COCA	The Cancer Genome Atlas Research Network (2012)
	PINS	Nguyen et al. (2017)
Kernel methods	rMKL-LPP	Speicher and Pfeifer (2015)
	mixKernel	Mariette and Villa-Vialaneix (2018)
	CIMLR	Ramazzotti et al. (2018)
	NEMO	Rappoport and Shamir (2019)
Latent variable models	iCluster	Shen, Olshen, and Ladanyi (2009)
	-	Zhang et al. (2012)
	LRAcluster	Wu et al. (2015)
Bayesian models	GIMMs	Liu et al. (2007)
	MDI	Kirk et al. (2012)
	BCC	Lock and Dunson (2013)
	Clusternomics	Gabašová, Reid, and Wernisch (2017)
	-	Afrin et al. (2020)
Graph-based methods	SNF	Wang et al. (2014)
	ANF	Ma and Zhang (2018)
	MONET	Rappoport, Safra, and Shamir (2020)
ML methods	Multimodal DBN	Liang et al. (2014)
	pathME	Lemsara, Ouadfel, and Fröhlich (2020)
	VAE for cancer data	Simidjievski et al. (2019)
Others	iPF	Kim et al. (2015)

TABLE 3.1: Summary of integrative clustering methods developed specifically for multi-omic data.

3.3.1 *Consensus clustering*

We recall here the main features of CC in order to be able to explain the functioning of COCA. As originally formulated, CC is an approach for assessing the robustness of the clustering structure present in a single dataset (Monti et al., 2003; Wilkerson and Hayes, 2010). The idea behind CC is that, by resampling multiple times the items that we want to cluster, and then applying the same clustering algorithm to each of the subsets of items, we assess the robustness of the clustering structure that the algorithm detects, both to perturbations of the data and (where relevant) to the stochasticity of the clustering algorithm. To do this, CC makes use of the concepts of co-clustering matrix and consensus matrix, which we recall below:

Definition 3.2 *Given a set of items $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ that we seek to cluster and a clustering $\boldsymbol{c} = [c_1, \ldots, c_N]$ such that c_i is the label of the cluster to which item $\boldsymbol{x}_i$ has been assigned, the corresponding* co-clustering matrix *(or* connectivity matrix*) is an $N \times N$ matrix C such that*

$$C_{ij} = \begin{cases} 1 & \text{if } c_i = c_j, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 3.3 *Let $X^{(1)}, \ldots, X^{(H)}$ be a list of perturbed datasets obtained by resampling subsets of items and/or covariates from the original dataset X. If $I^{(h)}$ is a subset of the indices of the observations $I = \{1, 2, \ldots, N\}$, and $X^{(h)}$ is the dataset containing only the statistical units corresponding to the indices in set I, then the co-clustering matrix has element (i, j) defined as follows:*

$$C_{ij}^{(h)} = \begin{cases} 1 & \text{if } i, j \in I \text{ and } c_i = c_j, \\ 0 & \text{otherwise.} \end{cases}$$

We denote by $C^{(h)}$ the co-clustering matrix corresponding to dataset $X^{(h)}$ where the items have been assigned to K classes using a clustering algorithm. The consensus matrix Δ^K *is an $N \times N$ matrix with elements*

$$\Delta_{ij}^K = \frac{\sum_{h=1}^H C_{ij}^{(h)}}{\min\left\{1, \sum_{h=1}^H \mathbb{1}_{ij}^{(h)}\right\}}$$

where $\mathbb{1}_{ij}^{(h)} = 1$ if both items i and j are present in dataset $X^{(h)}$.

Thus, CC performs multiple runs of a (stochastic) clustering algorithm (e.g. k-means, hierarchical clustering, etc.) to assess the stability of the discovered clusters, with the consensus matrix providing a convenient summary of the CC analysis. If all the elements of the consensus matrix are close to either one or zero, this means that every pair of items is either almost always assigned to the same

cluster, or almost always assigned to different clusters. Therefore, consensus matrices with all the elements close to either zero or one indicate stable clusters. The CC procedure for a fixed number of clusters K is reported in Algorithm 3.1.

Algorithm 3.1: Consensus clustering (CC).

Input : Dataset X, number of clusters K.
Initialise: Consensus matrix $\Delta^K = 0_{N\times N}$.
Matrix of resampling counts $D_{ij} = 0_{N\times N}$.

for $h \in \{1,\dots,H\}$ **do**
  $X^{(h)}$ = resample from the rows and/or columns of X
  $\mathbf{c}^{(h)}$ = divide the items of $X^{(h)}$ into K clusters
  $C^{(h)}$ = build the co-clustering matrix corresponding to $\mathbf{c}^{(h)}$
  for $i, j \in \{1,\dots,n\}$ **do**
    $\Delta^K_{ij} = \Delta^K_{ij} + C^{(h)}_{ij}$
    $D_{ij} = D_{ij} + \mathbb{1}^{(h)}_{ij}$
  end
end
for $i, j \in \{1,\dots,n\}$ **do**
  $\Delta^K_{ij} = \Delta^K_{ij} / \min\{D_{ij}, 1\}$
end

Output : Consensus matrix Δ^K.

In the framework of consensus clustering, these matrices can also be used to determine the number of clusters, by computing and comparing the consensus matrices Δ^K for a range of numbers of clusters $\mathcal{K} = \{K_{\min},\dots,K_{\max}\}$ of interest and then pick the value of K that gives the consensus matrix with the greater proportion of elements close to either zero or one (Monti et al., 2003).

3.3.2 *COCA algorithm*

In contrast to consensus clustering (which we emphasise is concerned with assessing clustering stability when analysing a single dataset), the main goal of COCA is to summarise the clusterings found in *different* 'omic datasets, by identifying a "global" clustering across the datasets that is intended to summarise the clustering structures identified in each of the individual datasets. In the first step, a clustering $\boldsymbol{c}^m$ is produced independently for each dataset X_m, $m = 1,\dots,M$, each with a different number of clusters K_m. We define $\bar{K} = \sum_{m=1}^{M} K_m$. Then, the clusters are summarised into a *matrix of clusters* (MOC) of size $\bar{K} \times N$, with elements

$$\mathrm{MOC}_{m_k,n} = \begin{cases} 1 & \text{if } c^m_n = m_k, \\ 0 & \textit{otherwise}. \end{cases}$$

where by m_k we denote the kth cluster in dataset m, $k = 1,\dots,K_m$ and $m = 1,\dots,M$. The MOC matrix is then used as input to CC (Algorithm 3.1) together

with a fixed global number of clusters K. The resulting consensus matrix computed with Algorithm 3.1 is then used as the similarity matrix for a hierarchical clustering method (or any other distance-based clustering algorithm). The procedure is summarised in Algorithm 3.2.

The global number of clusters K is not always known. In The Cancer Genome Atlas Research Network (2012), where COCA was introduced, the global number of clusters was chosen as in Monti et al. (2003), as explained above: CC was performed with different values of K and then the one that gave the "best" consensus matrices were considered. Instead, Aure et al. (2017) suggest to choose the value of K that maximises the average silhouette (see Rousseeuw, 1987, and Section 3.4.4) of the final clustering, since this was found to give more sensible results.

Algorithm 3.2: Cluster-of-clusters analysis (COCA).

Input : M datasets X_m, number of clusters K_m in each dataset, global number of clusters $\bar{K}$.

Initialise: MOC = $0_{K \times N}$.

for $m \in \{1, \dots, M\}$ **do**
  $\mathbf{c}^m$ = cluster the items in dataset X_m into K_m clusters
  for $n \in \{1, \dots, N\}$, $k \in \{1, \dots, K_m\}$ **do**
    Set $MOC_{n,m_k} = 1$ if $\mathbf{c}_i^m = k$
  end
end
for $h \in \{1, \dots, H\}$ **do**
  $MOC^{(h)}$ = resample from the rows and/or columns of MOC
  $\mathbf{c}^{(h)}$ = divide the items of $X^{(h)}$ into K clusters
  $C^{(h)}$ = build the co-clustering matrix corresponding to $\mathbf{c}^{(h)}$
  for $i, j \in \{1, \dots, n\}$ **do**
    $\Delta_{ij}^K = \Delta_{ij}^K + C_{ij}^{(h)}$
    $D_{ij} = D_{ij} + \mathbb{1}_{ij}^{(h)}$
  end
end
for $i, j \in \{1, \dots, n\}$ **do**
  $\Delta_{ij}^K = \Delta_{ij}^K / \min\{D_{ij}, 1\}$
end
Find final clustering $\boldsymbol{c}^{\bar{K}}$ using hierarchical clustering on $\Delta^{\bar{K}}$.

Output : Cluster labels $\boldsymbol{c}^{\bar{K}}$.

Because the construction of the MOC matrix just requires the cluster allocations, COCA has the advantage of allowing clusterings derived from different sources to be combined, even if the original datasets are unavailable or unwieldy. However, this method is unweighted, since all the clusters found in the first step have the same influence on the final clustering. Moreover, the objective function that is optimised by the algorithm is unclear.

In what follows, we describe an alternative way of performing integrative clustering, that takes into account not only the clusterings in each dataset, but also the information about the similarities between items that are extracted from different types of data. Additionally, the new method allows weights to be given to each source of information, according to how useful it is for defining the final clustering.

3.4 KERNEL LEARNING INTEGRATIVE CLUSTERING

Before introducing the new methodology, we recall the main principles behind the methods that we use to combine similarity matrices.

3.4.1 *Multiple kernel k-means clustering*

Gönen and Margolin (2014) extended the kernel k-means approach (Section 3.1) to the MKL setting. We consider multiple datasets $X_1, \ldots, X_M$ each with a different mapping function $\phi_m : \mathbb{R}^P \to \mathcal{H}_m$ and corresponding kernel $\delta_m(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi_m(\boldsymbol{x}_i), \phi_m(\boldsymbol{x}_j) \rangle_{\mathcal{H}_m}$ and kernel matrix Δ_m. Then, if we define

$$\boldsymbol{\phi}_\theta(\boldsymbol{x}_i) = [\theta_1 \phi_1(\boldsymbol{x}_i)', \theta_2 \phi_2(\boldsymbol{x}_i)', \ldots, \theta_M \phi_M(\boldsymbol{x}_i)']',$$

where $\boldsymbol{\theta} \in \mathbb{R}_+^M$ is a vector of kernel weights such that $\sum_m \theta_m = 1$ and $\theta_m \geq 0$, the kernel function of this multiple feature problem is a convex sum of the single kernels:

$$\begin{aligned} \delta_\theta(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \langle \boldsymbol{\phi}_\theta(\boldsymbol{x}_i), \boldsymbol{\phi}_\theta(\boldsymbol{x}_j) \rangle \\ &= \sum_{m=1}^{M} \langle \theta_m \phi_m(\boldsymbol{x}_i), \theta_m \phi_m(\boldsymbol{x}_j) \rangle_{\mathcal{H}_m} \\ &= \sum_{m=1}^{M} \theta_m^2 \delta_m(\boldsymbol{x}_i, \boldsymbol{x}_j). \end{aligned}$$

We denote the corresponding kernel matrix by Δ_θ. The optimisation problem now is

$$\begin{aligned} \underset{H, \boldsymbol{\theta}}{\text{maximise}} \quad & \operatorname{tr}(H' \Delta_{\boldsymbol{\theta}} H) - \operatorname{tr}(\Delta_{\boldsymbol{\theta}}) \\ \text{subject to} \quad & H'H = \mathbb{I}_K, \\ & \boldsymbol{\theta}' 1_M = 1, \\ & \Delta_{\boldsymbol{\theta}} = \sum_m \boldsymbol{\theta}_m^2 \Delta_m, \end{aligned}$$

where 1_M indicates a vector of ones of length M. The optimisation strategy proposed by Gönen and Margolin (2014) is based on the idea that, for some fixed

vector of weights $\boldsymbol{\theta}$, the problem is equivalent to the one of Equation (3.2a), where we had only one kernel. Therefore, they develop a two-step optimisation strategy: (i) given a fixed vector of weights $\boldsymbol{\theta}$, solve the optimisation problem as in the case of one kernel (Equation 3.3), with kernel matrix $\Delta_{\boldsymbol{\theta}}$ and then (ii) minimise the objective function with respect to the kernel weights, keeping the assignment variables fixed. This is a convex *quadratic programming* (QP) problem that can be solved with any standard QP solver up to a moderate number of kernels M. They also generalise this approach to a *localised* multiple kernel k-means, by assigning sample-specific weights, in order to remove sample-specific noise. This is achieved by defining a matrix of weights Θ, where each row corresponds to an observation and each column to one of the datasets. We indicate by θ_{im} the weight of observation $\boldsymbol{x}_i$ in dataset m and by $\boldsymbol{\theta}_m = [\theta_{1m}, \ldots, \theta_{Nm}]$ the vector of weights of dataset m. The mapping function is then $\boldsymbol{\phi}_\Theta(\boldsymbol{x}_i) = [\theta_{i1}\phi_1(\boldsymbol{x}_i)', \theta_{i2}\phi_2(\boldsymbol{x}_i)', \ldots, \theta_{iM}\phi_M(\boldsymbol{x}_i)']'$, and the corresponding kernel matrix Δ_Θ has element (i, j) defined as

$$\begin{aligned}
\delta_\Theta(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \langle \boldsymbol{\phi}_\Theta(\boldsymbol{x}_i), \boldsymbol{\phi}_\Theta(\boldsymbol{x}_j) \rangle \\
&= \sum_{m=1}^{M} \theta_{im}\theta_{jm} \langle \phi_{\Theta,m}(\boldsymbol{x}_i), \phi_{\Theta,m}(\boldsymbol{x}_j) \rangle_{\mathcal{H}_m} \\
&= \sum_{m=1}^{M} \theta_{im}\theta_{jm}\delta_m(\boldsymbol{x}_i, \boldsymbol{x}_j).
\end{aligned}$$

The optimisation problem in this case is analogous to the previous one:

$$\begin{aligned}
\underset{H,\Theta}{\text{maximise}} \quad & \operatorname{tr}(H'\Delta_\Theta H) - \operatorname{tr}(\Delta_\Theta) && \text{(3.8a)} \\
\text{subject to} \quad & H'H = 1_K, \\
& \Theta' 1_M = 1, \\
& \Delta_\Theta = \sum_m (\boldsymbol{\theta}_m \boldsymbol{\theta}_m') \circ \Delta_m,
\end{aligned}$$

where $\circ$ is the Hadamard product. Here too the objective function of Equation (3.8a) can be optimised using a two-step procedure, that iteratively (i) solves a standard kernel k-means problem with kernel Δ_Θ, keeping the weight matrix Θ fixed and then (ii) optimises the objective function with respect to Θ. Again, the first step reduces to solving one optimisation problem with a single kernel (Equations 3.3) and in the second step one just needs to solve a QP problem.

3.4.2 *Identifying consensus matrices as kernels*

Traditionally, the integration of multiple 'omic datasets via MKL has been made using closed form kernels. Lanckriet et al. (2004a), for instance, use the linear, diffusion, fast Fourier transform, and RBF kernels (Shawe-Taylor and Cristianini,

2004), among others, to perform ribosomal and membrane protein classification in yeast. Because it is not possible to know a priori which kernel is best, they use more than one kernel per data layer. Similarly, we have seen that Ramazzotti et al. (2018) generated a large number of RBF kernels for each data type, each with different parameters, and let the algorithm average over those. As our simulation studies show (Section 3.5), however, the choice of the kernel parameters is crucial.

Xing et al. (2003) developed an algorithm that learns the desired similarity between points in $\mathbb{R}^P$ when provided by the user with some examples of similar (or dissimilar) data points. This is known as *metric learning* (Kulis, 2013; Bellet, Habrard, and Sebban, 2013). Unfortunately this information is often not available in unsupervised situations. Therefore, how to best define kernels for each data layer remains an open question.

To address this, we prove that the consensus matrices defined in Section 3.3 are PSD, and hence that they can be used as input for any kernel-based clustering method, including the integrative clustering method presented in the next section.

Given any $N \times N$ co-clustering matrix C, we can reorder the rows and columns to obtain a block-diagonal matrix:

$$C = \begin{bmatrix} J_1 & 0 & 0 & \dots & 0 \\ 0 & J_2 & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & J_K \end{bmatrix}$$

where K is the total number of clusters and $J_k, k = 1, \dots, K$, is an $N_k \times N_k$ matrix of ones, with N_k being the number of items in cluster k. It is straightforward to show that the eigenvalues of a block diagonal matrix are simply the eigenvalues of its blocks. Since each block is a matrix of ones, the eigenvalues of each block are non-negative, and so any co-clustering matrix C is PSD. Moreover, given any set of λ_m, $m = 1, \dots, M$ non-negative, and co-clustering matrices C_m, $m = 1, \dots, M$, then $\sum_{m=1}^{M} \lambda_m C_m$ is PSD. Indeed, if λ is a non-negative scalar, and C is PSD, then λC is also PSD and the sum of PSD matrices is a PSD matrix. Since every consensus matrix is of the form $\sum_m \lambda_m C_m$, we can conclude that any consensus matrix is PSD.

In practice, when generating consensus matrices, computers often incur small round-off errors, which results in consensus matrices that are not PSD. Some of the most commonly used methods to convert similarity matrices into valid kernels are discussed by Chen et al. (2009). Among other things, they consider:

- to use *indefinite kernels*, simply ignoring the fact that Δ is indefinite;

- to perform a *spectrum clip*, i.e. given the eigenvalue decomposition $\Delta = U^T \Lambda U$, where Λ is the diagonal matrix $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N)$ and $\lambda_1, \ldots, \lambda_N$ are the eigenvalues of Δ, modify Λ as follows:

$$\Lambda_{\text{clip}} = \text{diag}(\max\{\lambda_1, 0\}, \ldots, \max\{\lambda_N, 0\});$$

- a *spectrum flip*:

$$\Lambda_{\text{flip}} = \text{diag}(|\lambda_1|, \ldots, |\lambda_N|);$$

- a *spectrum square*:

$$\Lambda_{\text{square}} = \Lambda^T \Lambda;$$

- or a *spectrum shift*:

$$\Lambda_{\text{shift}} = \Lambda + |\min\{\lambda_{\min}, 0\}| \mathbb{I}_N,$$

  where $\lambda_{\min}$ is the smallest eigenvalue of Δ, a is a constant such that $a \geq 1$ and $\mathbb{I}_N$ is the $N \times N$ identity matrix.

Here we use the spectrum shift, which is employed in many other applications. For example, in kernel ridge regression, shifting the eigenvalues of the kernel matrix corresponds to applying regularisation equivalent to that of Equation (2.2) (Shawe-Taylor and Cristianini, 2004, Chapter 3). In addition to that, in Section 3.5 we introduce cophenetic correlation coefficients, which we use to quantify how well defined is the clustering structure contained in a similarity matrix. It is interesting to note that the cophenetic correlation coefficient of a similarity matrix remains unchanged after a spectrum shift.

3.4.3 *KLIC algorithm*

We have shown above that any PSD matrix defines a feature map $\phi : \mathbb{R}^P \to \mathcal{H}$ and is therefore a valid kernel matrix. The integrative clustering method that we introduce here is based on the idea that we can identify the consensus matrices produced by Algorithm 3.1 as kernels. That is, one can perform consensus clustering on each dataset to produce a consensus matrix Δ_m for each $m \in \{1, \ldots, M\}$. This is a kernel Δ_m, where the ijth element corresponds to the similarity between items i and j. Therefore, these matrices Δ_m can be combined through the (localised) multiple kernel k-means algorithm described in Section 3.4.1. This allows a weight to be obtained for each kernel, as well as a global clustering $\boldsymbol{c}$ of the items. We refer to this as the KLIC (*kernel learning integrative clustering*) algorithm (see Figure 3.2 and Algorithm 3.3). We note that this algorithm could also be applied using more than one similarity matrix per dataset, and also using kernel matrices other than (or in addition to) consensus matrices.

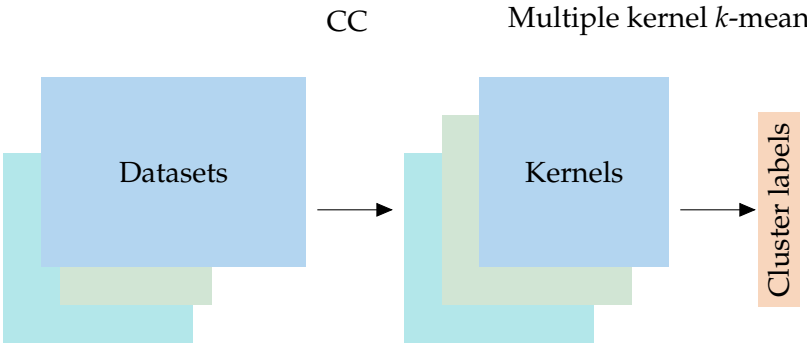


FIGURE 3.2: Schematic representation of KLIC. For each data layer, a kernel is formed using consensus clustering. All kernels are then combined in a weighted manner through localised multiple kernel k-means to obtain the final cluster labels.

Algorithm 3.3: Kernel learning integrative clustering (KLIC).

Input : M datasets X_m, maximum number of clusters K.

for $m \in \{1, \ldots, M\}$ **do**
  Δ_m = compute kernel for X_m
end
for $k \in \{1, \ldots, K\}$ **do**
  $[\boldsymbol{w}_k, \boldsymbol{c}_k]$ = apply multiple kernel k-means to $\Delta_1, \ldots, \Delta_M$
  $\boldsymbol{s}_k$ = calculate average silhouette of $\boldsymbol{c}_k$
end
Choose k such that $s_k \geq s_j, \forall j \neq k$.
return k, $\boldsymbol{w}_k$, $\boldsymbol{c}_k$.

Output : Best number of clusters k, set of kernel weights $\boldsymbol{w} = [w_1, \ldots, w_M]$, cluster labels $\boldsymbol{c} = [c_1, \ldots, c_N]$

3.4.4 *Choice of the number of clusters*

Many approaches exist in the literature to choose the number of clusters (see e.g. Milligan and Cooper, 1985; Tibshirani, 2001; Yeung, Haynor, and Ruzzo, 2001; Dudoit and Fridlyand, 2002; Kaufman and Rousseeuw, 1990). Here we restrict our attention to the metrics that have either been developed specifically for consensus/kernel matrices or that have been successfully used in contexts similar to the one considered here.

Area under the cumulative distribution function

For a given kernel matrix Δ, the empirical cumulative distribution function (CDF) is

$$\text{CDF}(x) = \frac{\sum_{i<j} \mathbb{1}(\Delta_{ij} < x)}{N(N-1)/2}$$

where $\mathbb{1}$ is the indicator function, Δ_{ij} denotes the ijth entry of the consensus/kernel matrix, and N is the number of rows and columns of Δ. Monti et al. (2003) suggest to use the change in the area under the CDF between two consecutive numbers of clusters to choose the best value of K. The area under the CDF corresponding to kernel $\Delta^{(K)}$ can be computed as

$$A(K) = \sum_{i=2}^{\tilde{N}} \left[x_{(i)} - x_{(i-1)} \right] \text{CDF}(x_{(i)})$$

where $\tilde{N} = N(N-1)/2$ and $x_{(1)}, \ldots, x_{(\tilde{N})}$ indicate the entries of the consensus matrix $\Delta^{(K)}$ sorted in increasing order. Monti et al. (2003) observed that, as K is increased the value of $A(K)$ markedly increases as long as $K < K_{\text{true}}$, where K_{true} is the true number of clusters. Instead, for $K > K_{\text{true}}$ the increase in area under the CDF is negligible. Therefore, one can choose the number of clusters to be the largest value of K for which the proportional increase in the area under the CDF, defined as

$$D(K) = \begin{cases} \text{A(K)} & \text{if K = 2,} \\ \text{A(K+1)/A(K)-1} & \text{if K > 2,} \end{cases}$$

is "large".

Proportion of ambiguous clustering

Şenbabaoğlu, Michailidis, and Li (2014) suggest to use a different measure based on the empirical CDF: the proportion of ambiguous clustering (PAC). This is defined as the fraction of sample pairs with consensus index values falling in the interval $(x_1, x_2) \in (0, 1)$, that is

$$\text{PAC}_K(x_1, x_2) = \text{CDF}_K(x_2) - \text{CDF}_K(x_1)$$

The number of clusters K corresponding to the minimum observed value of PAC_K is selected. However, they do not give any indication on how to choose the thresholds x_1 and x_2 except that they should be chosen near zero and one.

Silhouette

Another metric of cluster assessment that has been previously used in the context of integrative clustering (Wang et al., 2014) is the *silhouette*, a measure of the compactness of the clustering structure originally suggested by Rousseeuw (1987). There are two ways of defining the silhouette of a cluster, based respectively on the similarities and dissimilarities between the data. Here we briefly explain the former.

Given some cluster assignment labels $\boldsymbol{c} = [c_1, \dots, c_N]$ and some measure of the dissimilarity between the data points Δ_{ij} for all $i, j = 1, \dots, N$, we can define the following quantities: a_n is the average similarity of $\boldsymbol{x}_n$ to all the objects in cluster c_n and, for each $c_i \neq c_n$, Δ_{n,c_i} is the average similarity of n to all objects belonging to cluster c_i. Moreover, let us indicate by b_n the maximum Δ_{n,c_i} over all i such that $c_i \neq c_n$. Then, for each observation $n = 1, \dots, N$, we can calculate

$$s_n = \begin{cases} 1 - a_n/b_n, & \text{if } a_n < b_n, \\ 0, & \text{if } a_n = b_n, \\ a_n/b_n - 1, & \text{if } a_n > b_n. \end{cases}$$

This quantity takes values between -1 and 1, with higher values indicating that $\boldsymbol{x}_i$ has been allocated to an appropriate cluster and negative values suggesting that $\boldsymbol{x}_i$ has been misclassified.

The silhouette can be easily used in the context of MKL, defining the similarity between data points as the ijth entry of the kernel matrix Δ. Thus, we run our algorithms for combining the kernel matrices with different number of clusters from $K_{\min}$ to $K_{\max}$. We consider the overall average silhouette width $\bar{s} = \sum_{n=1}^{N} s_n$ as a measure of the compactness of clusters and we choose the value of K that gives the highest value of $\bar{s}$.

3.5 SIMULATION STUDY

To assess the KLIC algorithm described in Section 3.4 and to compare it to COCA, we perform a range of simulation studies.

We generate several synthetic datasets, each composed of data belonging to six different clusters of equal size. Each dataset has total number of observations equal to 300. Each observation $\boldsymbol{x}_n^{(k)}$ is generated from a bivariate normal with mean $k\tau$ for each variable, where k denotes the cluster to which the observation

belongs and τ the separation level of the dataset. Higher values of τ give clearer clustering structures. The variance covariance matrix is the identity matrix.

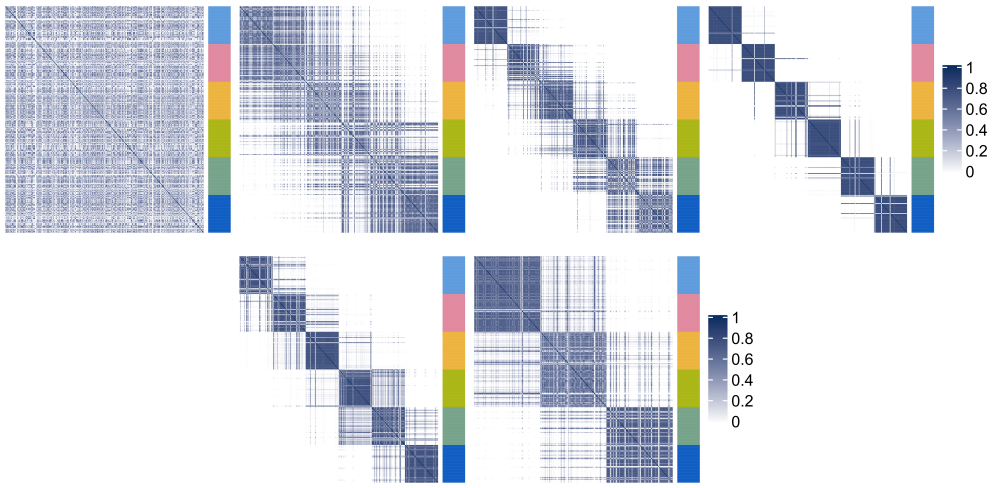


FIGURE 3.3: Consensus matrices of the synthetic data with different levels of noise going from "no cluster separability" to "high cluster separability". Blue indicates high similarity. The colours of the bar to the right of each matrix indicate the cluster labels.

We consider the following settings:

Setting A: similar datasets. We generate four datasets that have the same clustering structure and cluster separability τ. We denote the datasets by A, B, C, D. The goal of this experiment is to show that using localised kernel k-means on multiple consensus matrices leads to better results than those obtained using just one consensus matrix. We also repeat this experiment adding to each dataset 13 covariates that have no clustering structure, i.e.

$$\mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_{50}^{(k)} \sim \mathcal{N}([k\tau, k\tau, \underbrace{0, \ldots, 0}_{13}], \mathbb{I}_{15}), \quad \forall k = 1, \ldots, 6$$

where $\mathbb{I}_{15}$ is the 15×15 identity matrix.

Setting B: datasets with different levels of noise. In this case we consider four datasets that have the same clustering structure, but different levels of cluster separability τ. We denote the datasets by 0 for "no cluster separability", 1 "low cluster separability", 2 "medium cluster separability", and 3 "high cluster separability" (Figure 3.3). We use this example to show how the weights are allocated to each consensus matrix and why it is important to assign lower weights to datasets that are noisy or not relevant.

Setting C: datasets with nested clusters. We also investigate how the algorithm copes with the ambiguous situation of nested clusters. To this end, we generate two datasets with the same value of the parameter s setting the distance between cluster centres. The first one has six clusters, while the second one only has three clusters, each of them containing two of the clusters of the other dataset (Figure 3.3, second row).

We repeat each experiment 100 times. For each synthetic dataset, we use consensus clustering (Algorithm 3.1) to obtain the consensus matrices. For simplicity, we always let $K = 6$ in settings A and B, and either $K = 3$ or $K = 6$ in setting C, as specified below. As for the clustering algorithm, we use k-means clustering with Euclidean distance, which we found to work well in practice. Appendix B contains additional simulation settings. In particular, we consider a wide range of separability values for the setting with four similar datasets. Moreover, we perform a short sensitivity analysis of the choice or tuning options for the k-means algorithm.

In the remainder of this section, we first apply the developed methods to the synthetic datasets and then compare the performances of our method for integrative clustering to COCA and other competitor methods.

3.5.1 *Assessment of KLIC*

We apply KLIC (Algorithm 3.3) to the synthetic datasets generated for settings A, B, and C.

Setting A: similar datasets. First, we run the kernel k-means algorithm on each of the consensus matrices that have the same clustering structure and noise level. Then, we use Algorithm 3.3 to run KLIC on multiple datasets.

We use the adjusted Rand index (ARI) of Hubert and Arabie (1985) as a measure of the similarity between the output of each clustering method and the true partition of the data. To compute the ARI, given two partitions $U = \{U_1, \ldots, U_K\}$ and $V = \{V_1, \ldots, V_L\}$, we summarise the overlapping between each pair of subsets U_i and V_j in a contingency table where $\nu_{ij} = |U_i \cap V_j|$ (Table 3.2).

Class	v_1	v_2	$\ldots$	v_L	Sums
u_1	ν_{11}	ν_{12}	$\ldots$	ν_{1L}	$\nu_{1\cdot}$
u_2	ν_{21}	ν_{22}	$\ldots$	ν_{2L}	$\nu_{2\cdot}$
$\vdots$	$\vdots$			$\vdots$	$\vdots$
u_k	ν_{k1}	ν_{k2}	$\ldots$	ν_{KL}	$\nu_{K\cdot}$
Sums	$\nu_{\cdot 1}$	$\nu_{\cdot 2}$	$\ldots$	$\nu_{\cdot L}$	N

TABLE 3.2: Contingency table of two partitions of the data.

The ARI is then calculated as

$$ARI = \frac{\sum_{k,l}\binom{v_{kl}}{2} - \sum_k \binom{v_{k\cdot}}{2}\sum_l \binom{v_{\cdot l}}{2}/\binom{v}{2}}{\frac{1}{2}\left[\sum_k \binom{v_{k\cdot}}{2} + \sum_l \binom{v_{\cdot l}}{2}\right] - \sum_k \binom{v_{k\cdot}}{2}\sum_l \binom{v_{\cdot l}}{2}/\frac{v}{2}}.$$

This is the corrected-for-chance version of the Rand index (Rand, 1971) that is simply the number n_s of pairs of elements that are in the same subsets in both partitions U and V, plus the number n_d of pairs of elements that are in different subsets in both U and V, divided by the total number of pairs N^2:

$$RI = \frac{n_s + n_d}{N^2}.$$

In Figure 3.4a are reported the box plots of the ARI obtained combining the four datasets together using KLIC (column "A+B+C+D"). Figure 3.4b shows the box plots of the average weights assigned by the KLIC algorithm to the observations in each dataset.

We observe that, as expected, combining together more datasets enables the clustering structure to be more accurately recovered than just taking the matrices one at a time. This is because localised kernel k-means allows to give different weights to each observation. Therefore, if data point n is hard to classify in dataset d_1, but not in dataset d_2, we will have $\theta_{nd_1} < \theta_{nd_2}$. However, on average the weights are divided equally between the datasets. This reflects the fact that all datasets have the same dispersion and, as a consequence, they contain on average the same amount of information about the clustering structure.

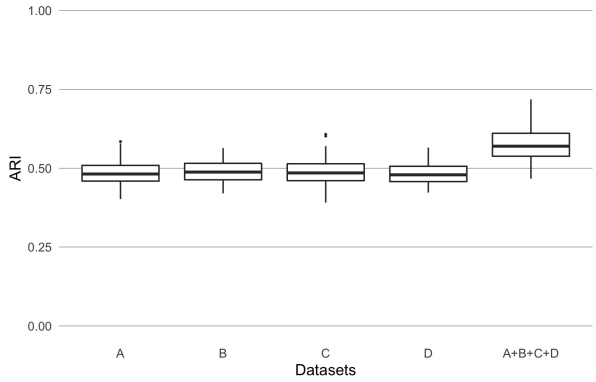
Setting B: datasets with different levels of noise. Here we use the datasets shown in Figure 3.3, that have the same clustering structure (six clusters of the same size each) but different levels of cluster separability. We consider four different settings, each time combining three out of the four synthetic datasets. Figure 3.5a shows the box plots of the ARI obtained using kernel k-means on the datasets taken one at a time (columns "0", "1", "2", "3") and the ARI obtained using KLIC on each subset of datasets (columns "0+1+2", "0+1+3", "0+2+3", "1+2+3"). As expected, the consensus matrices with clearer clustering structure give higher values of the ARI on average. Moreover, the ARI obtained combining three matrices with different levels of cluster separability is on average the same or higher as in the case when only the "best" matrix is considered. This is because larger weights are assigned to the datasets that have clearer clustering structure. In Figure 3.5b are reported the box plots of the average weights given by the localised multiple kernel k-means to the observations in each dataset. It is easy to see that each time the matrix with best cluster separability has higher weights than the other two.

Setting C: datasets with nested clusters. For the CC step, we use the true number of clusters for each dataset. However, since the localised kernel k-means algorithm works only with a fixed number of clusters, we try both with $K = 3$ and $K = 6$. The ARI and the average weights assigned to each matrix are reported in Figure 3.6. For $K = 6$, the weights assigned to each matrix are not as we expected: the matrix with three clusters is weighted slightly more highly than the other one. To investigate this phenomenon, we introduce an additional way to score how strong the signal is in each dataset. We use the *cophenetic correlation coefficient*, a measure of how faithfully hierarchical clustering would preserve the pairwise distances between the original data points (Sokal and Rohlf, 1962; Brunet et al., 2004). Given a dataset $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ and a similarity matrix $\Delta \in \mathbb{R}^{N \times N}$, we define the *dendrogrammatic distance* between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as the height of dendrogram at which these two points are first joined together by hierarchical clustering and we denote it by η_{ij}. The cophenetic correlation coefficient ρ is calculated as

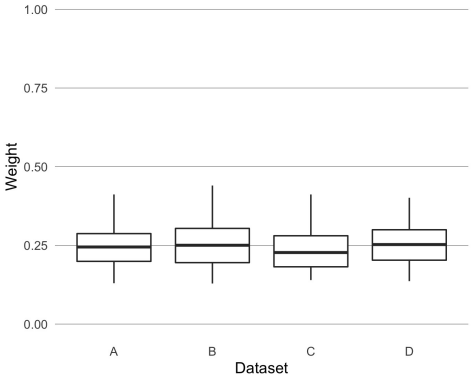
$$\rho = \frac{\Sigma_{i<j}(\Delta_{ij} - \bar{\Delta})(\eta_{ij} - \bar{\eta})}{\sqrt{\Sigma_{i<j}(\Delta_{ij} - \bar{\Delta})\Sigma_{i<j}(\eta_{ij} - \bar{\eta})}},$$

where $\bar{\Delta}$ and $\bar{\eta}$ are the average values of Δ_{ij} and η_{ij} respectively. The cophenetic correlation coefficient of a consensus matrix can be interpreted as an indication of the level of its dispersion or, equivalently, of the stability of the clustering used in CC. If the clusters are invariant under subsampling of the data features/observations, then the consensus matrix has all entries equal to either one or zero, and cophenetic correlation coefficient equal to one. On the other hand, if clusters vary at each iteration of consensus clustering, the entries of the consensus matrix are scattered between zero and one, and the corresponding cophenetic correlation coefficient is negative. The consensus matrices shown in Figure 3.3, for instance, have increasing cophenetic correlation going from left (lower cluster separability) to right (higher cluster separability). We find that in this case the consensus matrices with $K = 3$ have slightly higher cophenetic correlation than the ones with $K = 6$ with the same level of cluster separability s. This explains why higher weights are assigned to the former. This suggests that, in ambiguous cases, localised kernel k-means assigns higher weights (on average) to the kernels with highest cophenetic correlation. Intuitively, the sum of within-cluster distances in the feature space is zero when each pair of data points has similarity one if both data points are in the same cluster, and zero otherwise. Minimising that sum thus corresponds to finding the weights and cluster allocations that lead to a weighted kernel that is as close as possible to a kernel with cophenetic correlation one.

We also report the results obtained setting either $K = 3$ or $K = 6$ at each step of KLIC, i.e. consensus clustering of each dataset and MKL (Figure 3.7).

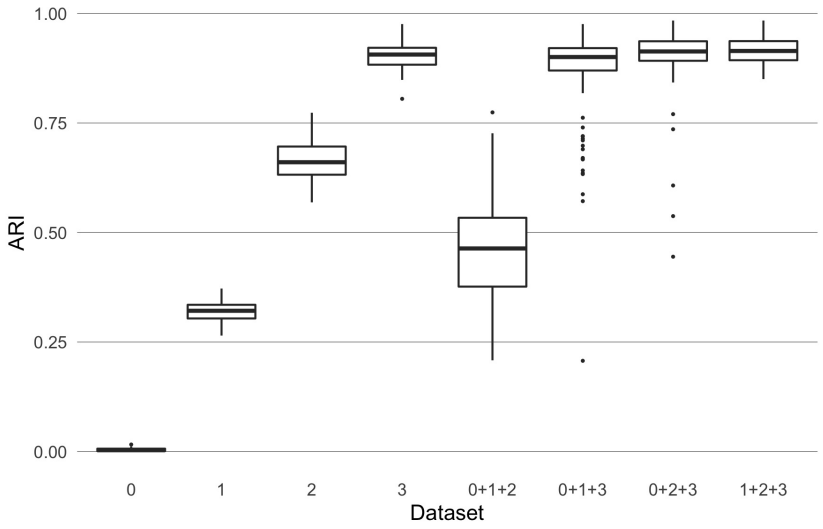


(A) Adjusted Rand index

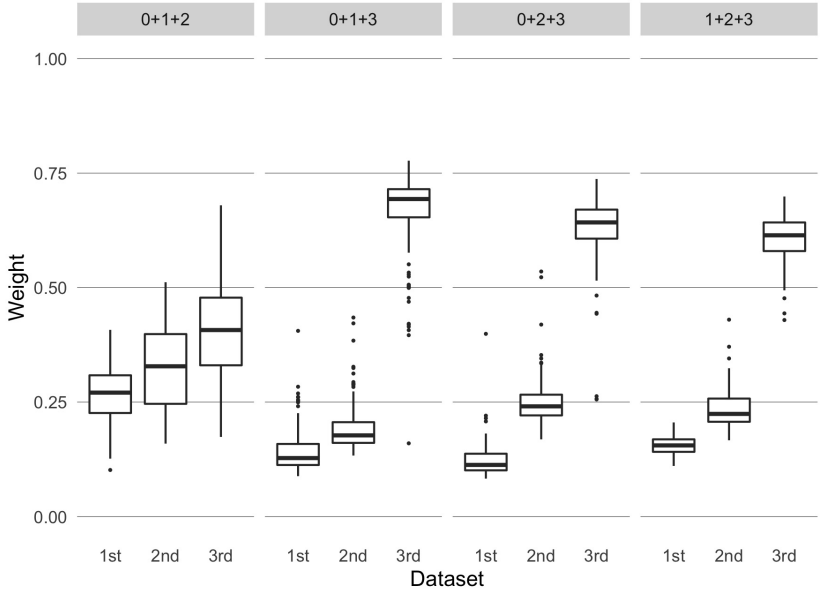


(B) Weights

FIGURE 3.4: Results of applying KLIC to four similar datasets. On the left is the ARI of KLIC applied to each dataset separately (columns "A", "B", "C", and "D") and to all four datasets together (column "A+B+C+D"). The ARI is higher in the last column because KLIC can combine information from all the datasets to find a global clustering. On the right are the kernel weights associated to each dataset, when applying KLIC to all four datasets together. The algorithm is able to recognise that each dataset contains the same amount of information regarding the global clustering, and therefore assigns on average the same weight to each dataset.

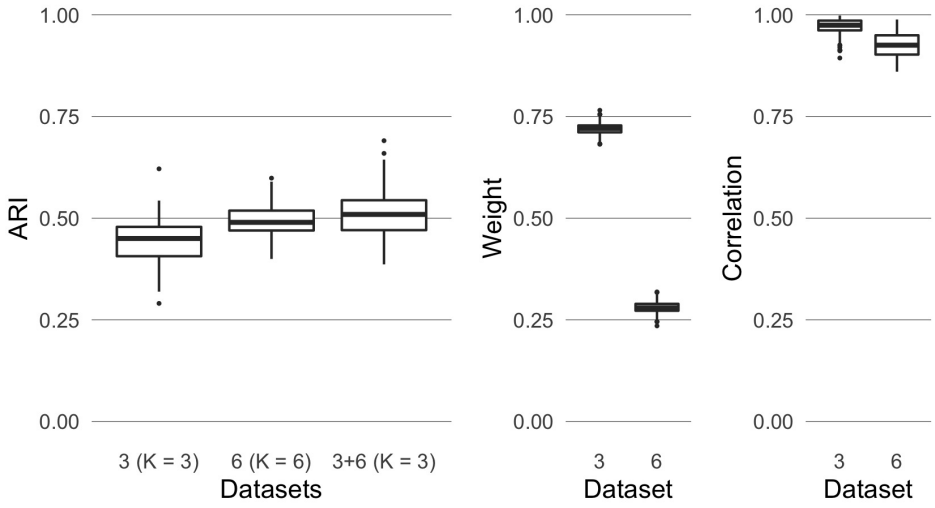


(A) Adjusted Rand index.

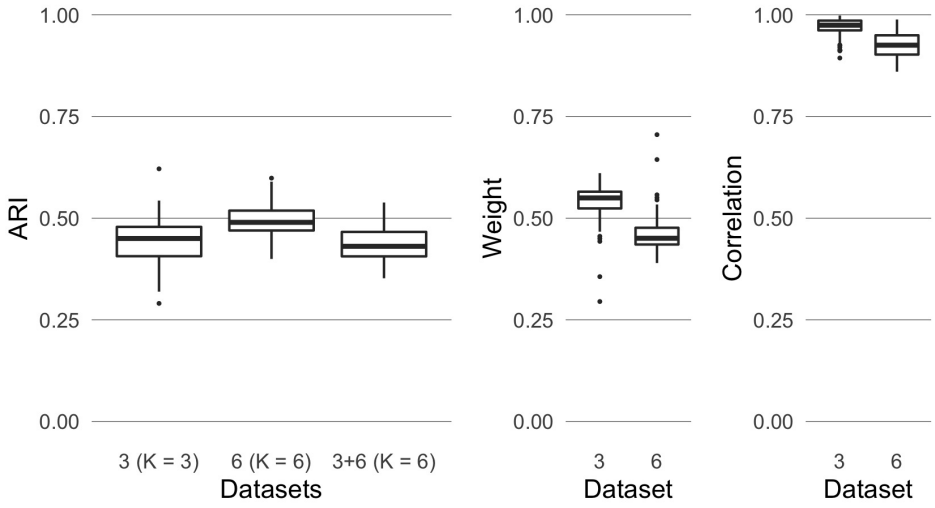


(B) Weights.

FIGURE 3.5: Results of applying KLIC to datasets with different levels of noise ("0" indicates the dataset that has no cluster separability, "1" the dataset with low cluster separability, and so on). (A) ARI of KLIC applied to each dataset separately (columns "0", "1", "2", and "3") and to subsets of three of those datasets (columns "0+1+2", "0+1+3", "0+2+3", and "1+2+3"). (B) Kernel weights associated to each dataset in each of the experiments with multiple datasets, ordered by cluster separability. For example, the first subset is "0+1+2" so the weights marked as "1st" are those assigned to dataset "0", "2nd" are those assigned to "1" and so on. For each subset of datasets the weights of the noisier datasets ("1st" and "2nd") are lower than those of the "best" dataset in the subset ("3rd"). This is reflected in an increased ARI in each subset, compared to applying KLIC to those datasets separately.

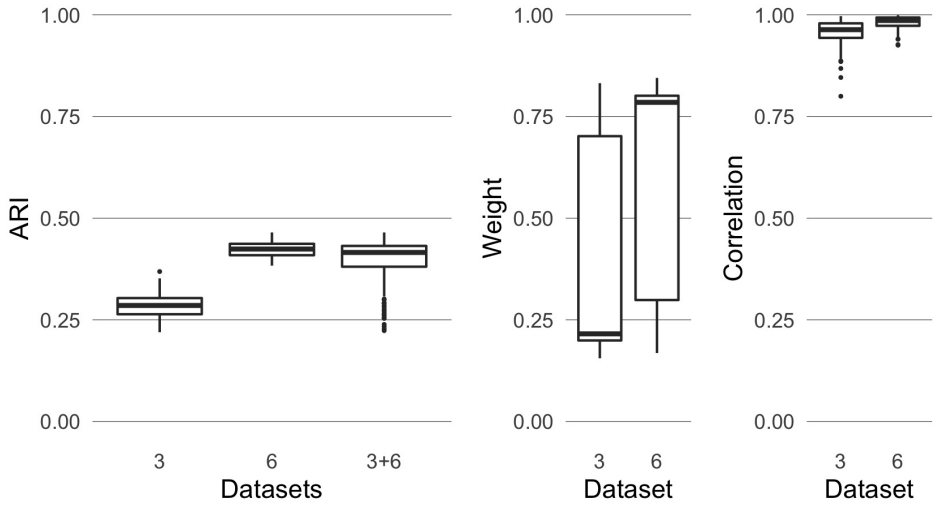


(A) True number of clusters for CC, $K = 3$ for global clustering.

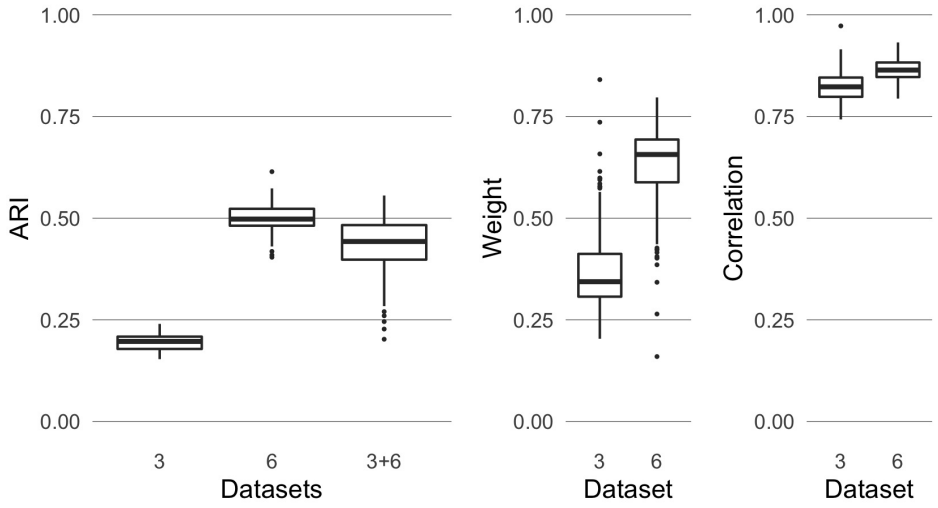


(B) True number of clusters for CC, $K = 6$ for global clustering.

FIGURE 3.6: Results of applying KLIC to datasets that have nested clusters. Left: ARI of KLIC applied to the datasets with three and six clusters separately (columns "3" and "6" respectively) and to those two datasets combined (column "3+6"). Centre: the weights assigned to each dataset. Right: cophenetic correlation coefficients of the consensus matrices built with $K = 3$ (for the dataset with three clusters) and $K = 6$ (for the dataset with six clusters). Higher weights are given to the kernels with higher cophenetic correlation, irrespectively of their number of clusters.



(A) $K = 3$ at each step.



(B) $K = 6$ at each step.

FIGURE 3.7: Results of applying KLIC to datasets that have nested clusters. Left: ARI of KLIC applied to the datasets with three and six clusters separately (columns "3" and "6" respectively) and to those two datasets combined (column "3+6"). Centre: the weights assigned to each dataset. Right: cophenetic correlation coefficients of the consensus matrices built with $K = 3$ (top) and $K = 6$ (bottom). Higher weights are given to the kernels with higher cophenetic correlation, irrespectively of their number of clusters.

3.5.2 *Comparison between KLIC, COCA, and other methods*

We compare the performance of KLIC to the one obtained using COCA, as well as to two other comparable integrative clustering algorithms for which implementations are readily available; namely, iCluster and Clusternomics. Additionally, we compare to localised multiple kernel k-means using standard RBF kernels. We use the same synthetic datasets as in the previous section.

For COCA, we use the k-means algorithm with Euclidean distance, fixing the number of clusters to be equal to the true one, to find the clustering labels of each dataset. Many other clustering algorithms can be used, but we found that this is the one that gives the best results among the most common ones. To find the global clustering, we build the consensus matrices using 1000 resamplings of the data, each time with 80% of the observations and all the features. The final clustering is done using hierarchical clustering with average linkage on the consensus matrix. The iCluster model is fitted using the `tune.iCluster2` function of the R package `iCluster` (Shen, 2012), which helps selecting the parameters for the penalty terms, with number of clusters set to six. For Clusternomics we use the `contextCluster` function of the R pacakge `clusternomics` (Gabašová, Reid, and Wernisch, 2017), providing the true number of clusters both for the partial and global clusterings.

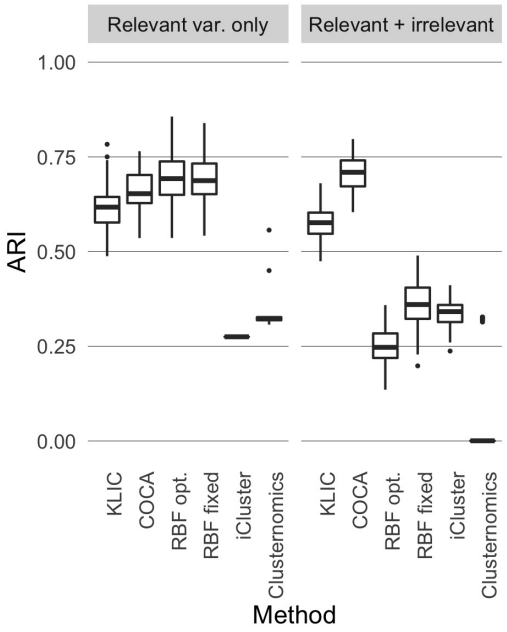
To assess the impact of the choice of the RBF kernel parameters on the final clustering, we consider two ways to set the free parameter of the RBF kernel. In one setting we fix $\sigma = 1$, a common default value. In the second setting, σ is tuned for each dataset to maximise the average ARI between the clustering obtained with kernel k-means on the RBF kernel and the true clusters (more information about this procedure can be found in Appendix B). Although this procedure clearly could not be applied in practice (where the true clustering is unknown), it is used here to determine a putative upper bound on the performances of MKL with this kernel.

Setting A: similar datasets. We combine four datasets that have the same clustering structure and cluster separability. In Figure 3.8a is shown the ARI of all considered methods applied to 100 sets of data of this type. In the first setting, where only variables relevant for the clustering are present, the localised multiple kernel k-means with RBF kernel has the highest median ARI, followed by COCA and KLIC. To cluster the data that include noisy variables, we replace the k-means algorithm by the sparse k-means feature selection framework of Witten and Tibshirani (2010) in COCA and KLIC, using the R package `sparcl` (Witten and Tibshirani, 2018). Thanks to this, the performances of these two methods are not affected by the presence of irrelevant variables. COCA, in particular, has the highest median ARI, followed by KLIC. This shows that both methods work

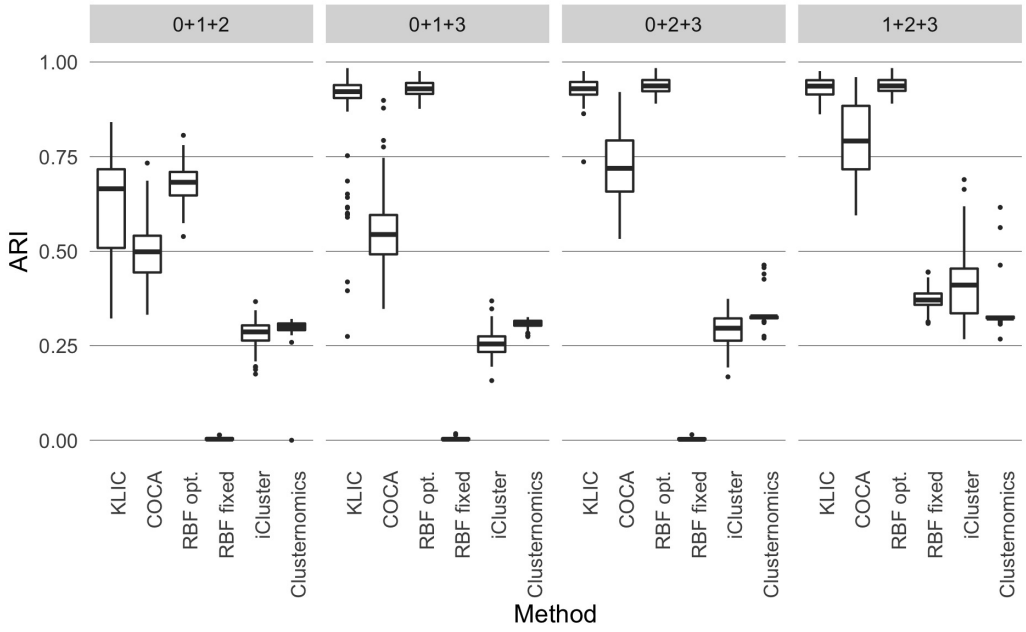
well in the case of multiple datasets that have the same clustering structure and level of noise and, in contrast to the four other methods considered here, can be straightforwardly modified to deal with the presence of irrelevant features.

Setting B: datasets with different levels of noise. We also compare the behaviour of all methods in the presence of multiple datasets with the same clustering structure, but different levels of cluster separability. The ARI is shown in Figure 3.8b. We observe that, in each of the four simulation settings, KLIC and the optimised version of localised multiple kernel k-means with RBF kernel have the highest ARI scores. The reason for this is that COCA, iCluster, and Clusternomics are not weighted methods, so their ability to recover the true clustering structure is decreased by adding noisy datasets. Instead, we have shown in the previous section that KLIC allows to give lower weights to the noisiest datasets, achieving better performances. We emphasise that the optimal values of the RBF parameters have been determined making use of the true cluster labels, which is not possible in real applications. The performance achieved when the RBF kernel parameter, σ, is fixed to 1 may be therefore be more representative of what can be achieved in practice.

Overall, these comparisons suggest that KLIC may be a good default choice, since it can be run in such a way that it is robust to both the inclusion of noisy variables (via the choice of an appropriate clustering algorithm) and of noisy datasets.



(A) Similar datasets.



(B) Datasets with different level of noise.

FIGURE 3.8: Comparison between KLIC, COCA, and other integrative clustering algorithms. The labels "RBF opt." and "RBF fixed" refer to the MKL method using an RBF kernel with either σ optimised or dixed at 1 (see text). (A) ARI obtained with each clustering algorithm using four datasets having the same clustering structure and cluster separability (as in Figure 3.4). (B) ARI obtained with each clustering algorithm for each of the subsets of heterogeneous datasets considered in Figure 3.5. The high ARI obtained with KLIC in all settings shows the advantage of using this method, especially when some of the datasets are noisy.

3.6 MULTIPLATFORM ANALYSIS OF 12 CANCER TYPES

We consider here the multiplatform integrative analysis of 3,527 tumour samples performed by Hoadley et al. (2014) that was mentioned in the Introduction. In this study, 11 tumour subtypes were identified using COCA, from samples of 12 different tumour types glioblastoma multiforme (GBM), serous ovarian carcinoma (OV), colon (COAD) and rectal (READ) adenocarcinomas, lung squamous cell carcinoma (LUSC), lung adenocarcinoma (LUAD), breast cancer (BRCA), acute myelogenous leukemia (AML), endometrial cancer (UCEC), renal cell carcinoma (KIRC), bladder urothelial adenocarcinoma (BLCA), and head and neck squamous cell adenocarcinoma (HNSC). To do so, they applied different clustering algorithms to each data type separately: DNA copy number, DNA methylation, mRNA expression, miRNA expression, and protein expression. They then combined the five sets of clusters obtained in this way using COCA. The final clusters are highly correlated with the tissue-of-origin of each tumour sample, but some cancer types coalesce into the same clusters. The clusters obtained in this way were shown to be prognostic and to give independent information from the tissue-of-origin.

Here, we use the same data to try to replicate their analysis, and compare the clusters obtained with COCA to those obtained with KLIC. To facilitate future analyses by other researchers, we have made available our scripts for processing and analysing these datasets using R, which include scripts that seek to replicate the original analysis of Hoadley et al. (2014), at `https://github.com/acabassi/klic-pancancer-analysis`.

In order to replicate the analysis performed by Hoadley et al. (2014), we preprocessed the DNA copy number, DNA methylation, mRNA expression, miRNA expression, and protein expression data in the same way as Hoadley et al. (2014) did. We then clustered the tumour samples independently for each dataset, using the same clustering algorithm as in the original paper. We compared the clusters we obtained to those reported by Hoadley et al. (2014) for different number of clusters, and we found that the best correspondence was given by choosing the same number of clusters as in the original paper, except for the miRNA expression data, for which we found the best number of clusters to be seven (instead of 15). Figure 3.9 shows the MOC matrix formed by these clusters and the resulting COCA clusters. As can be seen from the Figure, each dataset has some missing observations. The corresponding entries in the MOC matrix were set to zero. We chose the number of clusters that maximises the silhouette, as suggested by Aure et al. (2017), which is ten.

We then applied KLIC to the preprocessed data, building one consensus matrix for each dataset, using the same clustering algorithm and number of clusters as for COCA, and combining them as described in Algorithm 3.3. We assigned

weight zero to every missing observation, as explained in Section 5.1.1. The weighted consensus matrix is shown in Figure 3.10. The weights assigned on average to the observations in each dataset are as follows: copy number 31.4%, methylation 19.2%, miRNA 17.8%, mRNA 16.4%, protein 15.2%.

Similarly to what was observed by Hoadley et al. (2014), both the clusters obtained using COCA and KLIC correspond well with the tissue-of-origin classification of the tumours. However, there are a few differences between the two: the coincidence matrix is shown in Figure 3.11. Further details on how we tried to replicate the data analysis of Hoadley et al. (2014) and how we applied KLIC to these data can be found in Appendix B.

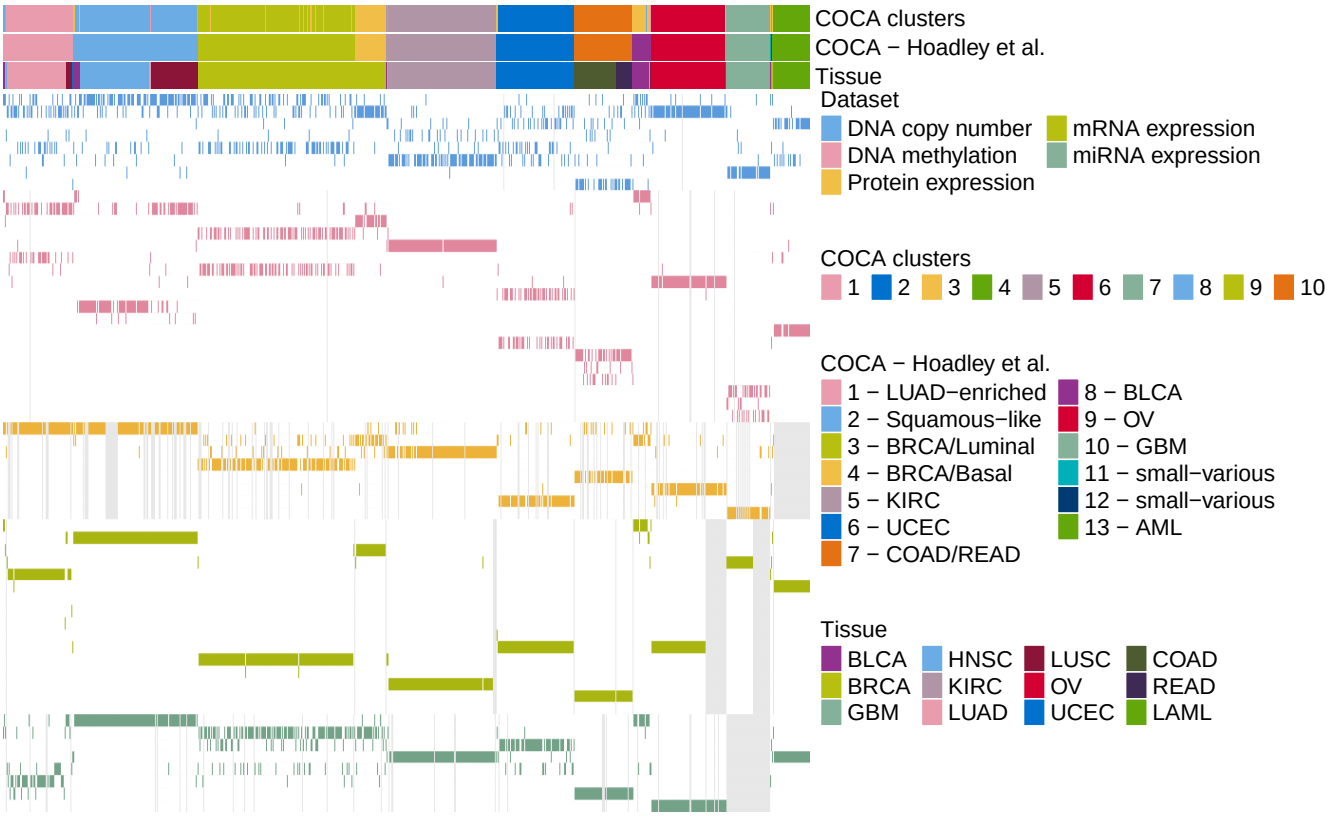


FIGURE 3.9: Multiplatform analysis of 12 cancer types. Matrix of clusters of the pan-cancer data: each row corresponds to a cluster in one of the dataset, and each column corresponds to a tumour sample. Coloured cells show which tumours belong to each cluster. Gray cells indicate missing observations.

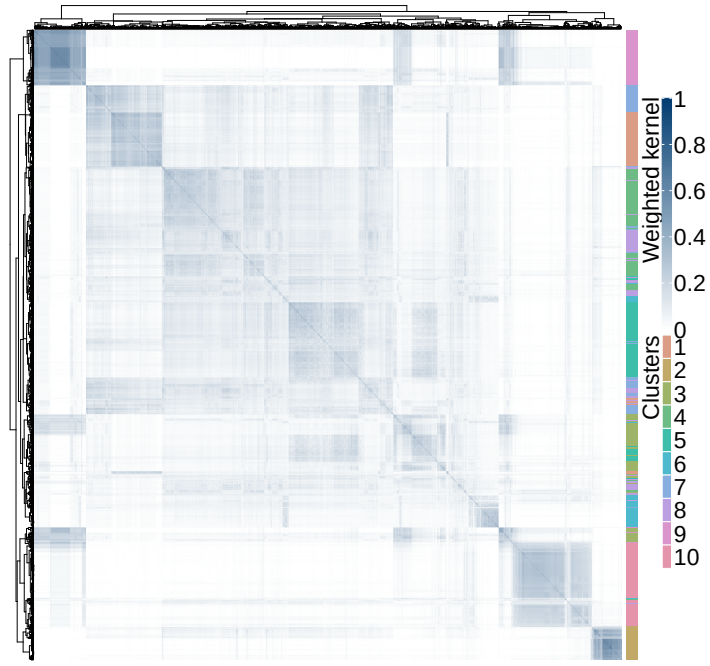


FIGURE 3.10: Multiplatform analysis of 12 cancer types. Left: weighted similarity matrix, where the rows and columns correspond to cancer samples. Higher values of similarity between samples are indicated in blue. Right: final clusters obtained using KLIC.

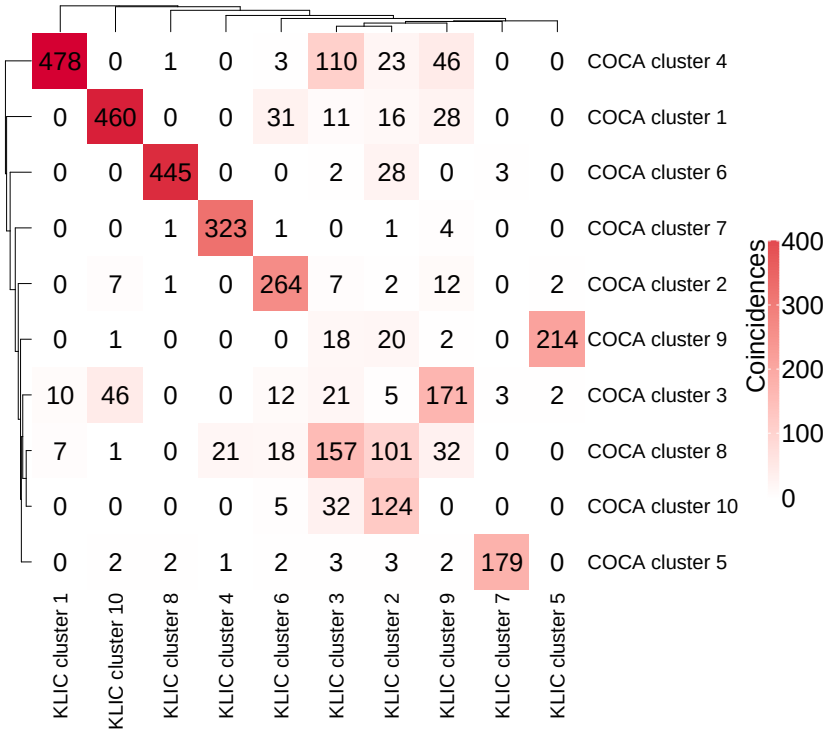


FIGURE 3.11: Multiplatform analysis of 12 cancer types. Coincidence matrix comparing the clusters given by COCA (rows) and KLIC (columns).

3.7 TRANSCRIPTIONAL MODULE DISCOVERY

Recall from the Introduction that transcriptional modules are groups (i.e. clusters) of genes that share a common biological function and are co-regulated by a common set of transcription factors. It has been recognised that integrative clustering methods can be useful for discovering transcriptional modules, by combining gene expression datasets with datasets that provide information about transcription factor binding (Ihmels et al., 2002; Savage et al., 2010).

Here we consider transcriptional module discovery for yeast (*Saccharomyces cerevisiae*). We integrate the expression dataset of Granovskaia et al. (2010), which contains measurements related to 551 genes whose expression profiles have been measured at 41 different time points of the cell cycle, with the ChIP-chip dataset of Harbison et al. (2004), which provides binding information for 117 transcriptional regulators for the same genes. The latter was discretised as in Savage et al. (2010) and Kirk et al. (2012).

We clustered the 551 genes based on the gene expression and transcription factor data using KLIC. For each dataset, the consensus matrices were obtained as explained in Section 3.3. The clustering algorithms used in this step were *partitioning around medoids* (PAM) with the correlations between data points as distances for the gene expression data (Kaufman and Rousseeuw, 1990) and *Bayesian hierarchical clustering* (BHC) for the transcription factor data (Heller and Ghahramani, 2005; Cooke et al., 2011). BHC is a fast approximate inference method for a class of model-based clustering techniques called *Dirichlet process mixture models*, which are introduced in Chapter 4. As we discuss in Chapter 4, one of the advantages of these methods is that they do not require the number of clusters to be set by the user. The consensus matrices were then used as input to KLIC. The algorithm was run with number of clusters ranging from 2 to 20. We found that the silhouette is maximised by setting the number of clusters to four. Figure 3.12 shows the weighted kernel matrix given by KLIC where the rows and columns are sorted by final cluster. Next to it are reported the data, where the observations are in the same order as in the kernel matrix. The clusters obtained independently on each dataset are also shown on the right of each plot. The kernel matrices of each dataset and corresponding weights and cophenetic correlation coefficients can be found in Appendix B.

We also applied COCA to this dataset, with the initial clusters for each dataset obtained with the same clustering algorithms as those used for the consensus matrices. The metrics used to choose the number of clusters for the initial clustering of the expression data are reported in Appendix B. For the final clustering the number of clusters was chosen in order to maximise the silhouette, considering all values between two and ten. This resulted in choosing the 10-cluster solution.

In order to assess the quality of the clusters, we make use of the gene ontology term overlap (GOTO) scores of Mistry and Pavlidis (2008). Each score is an indication of the number of annotations that, on average, are shared by genes belonging to the same clusters. These are available for three different ontologies: biological process (BP), molecular function (MF) and cell component (CE). More formally, denoting by $\text{annot}g_i$ the set of all direct annotations for each gene and all of their associated parent terms, the GOTO similarity between two genes g_i, g_j is the number of annotations that the two genes share:

$$\text{sim}_{\text{GOTO}}(g_i, g_j) = |\text{annot}g_i \cap \text{annot}g_j|.$$

To assess the quality of clusters, the overall GOTO scores associated with each of the biological process, molecular function, and cellular component ontologies of each clustering are used. Defining the mean GOTO score of each (non-singleton) cluster k as

$$\overline{\text{GOTO}}(k) = \frac{2}{N_k(N_k-1)} \sum_{g_i,g_j \in k} \text{GOTO}(g_i, g_j),$$

where N_k is the number of genes in cluster k, the overall GOTO score is defined as the weighted average of the mean GOTO scores

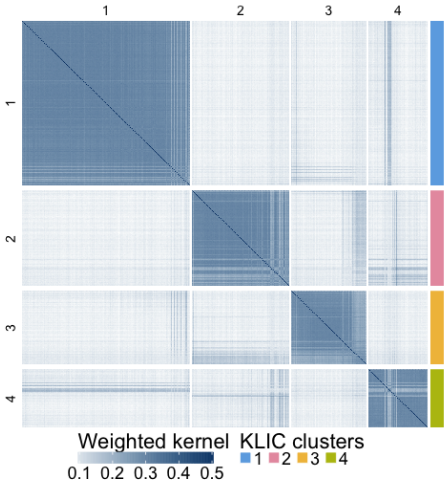
$$\overline{\text{GOTO}}_{\text{overall}} = \sum_{k=1}^{\tilde{K}} \left[\left(\frac{N_k}{\tilde{N}} \right) \overline{\text{GOTO}}(k) \right],$$

where K is the total number of non-singleton clusters and $N = \sum_{k=1}^{K} N_k$.

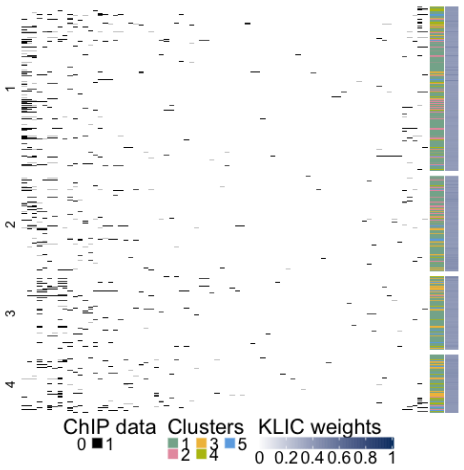
We report in Table 3.3 the GOTO scores of both KLIC and COCA clusters, for both number of clusters selected by KLIC (four) and COCA (ten). We also show the scores obtained clustering each dataset separately. We observe that, while in the case of four clusters no information is lost by combining the datasets, by dividing data into ten clusters one obtains more biologically meaningful clusters. Moreover, KLIC does a better job at combining the datasets, by better exploiting the information contained in the data and down-weighting the kernel of the ChIP dataset, which contains less information. More details about the kernel matrices and weights can be found in Appendix B.

3.8 DISCUSSION

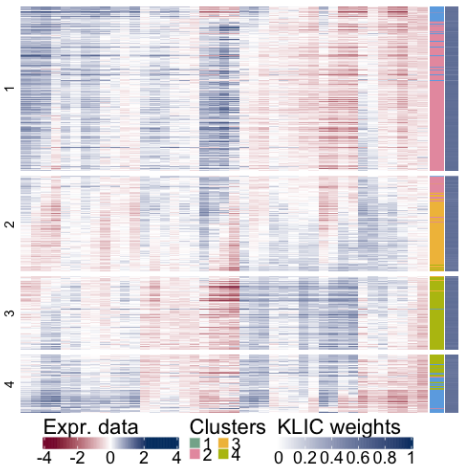
In this chapter, the focus has shifted from supervised to unsupervised integration of multi-omic data. Some of the challenges encountered here (e.g. the large number of missing values, the high dimensionality of the data) are the same as in Chapter 2. These were taken into account and addressed in this chapter. However, this work has also brought to light some difficulties that are specific to cluster analysis, which are summarised here.



(A) Weighted kernel matrix.



(B) ChIP data.



(C) Expression data.

FIGURE 3.12: Transcriptional module discovery, KLIC output. (A) Weighted kernel matrix obtained with KLIC, where each row and column corresponds to a gene, and final clusters. (B) Transcription factor data, where each row represents a gene and each column a transcription factor, black dots correspond to transcription factors that are believed to be able to bind to the promoter region of the corresponding gene with high confidence; clusters obtained using BHC on the transcription factor data and weight assigned by KLIC to each data point. (C) Gene expression data, where each row is a gene and each column a time point, clusters obtained using PAM on the gene expression data, and weights assigned by KLIC to each data point.

Clusters	Dataset(s)	Algorithm	GOTO BP	GOTO MF	GOTO CE
8	ChIP	BHC	6.09	0.90	8.33
4	Expression	PAM	6.12	0.91	8.41
4	ChIP+Expression	COCA	6.12	0.91	8.41
4	ChIP+Expression	KLIC	6.12	0.91	8.41
10	ChIP+Expression	COCA	6.28	0.93	8.51
10	ChIP+Expression	KLIC	**6.32**	**0.95**	**8.53**

TABLE 3.3: Gene ontology term overlap scores for different sets of data, clustering algorithms and numbers of clusters. BP stands for biological process ontology, MF for molecular function, and CE for cell component.

3.8.1 *Main findings*

In the first part of the chapter we have given the algorithm for COCA, a widely used method in integrative clustering of genomic data, highlighting the main issues of using this method. We have also presented KLIC, a novel approach to integrative clustering, that allows multiple datasets to be combined to find a global clustering of the data and is well-suited for the analysis of large datasets, such as those often encountered in genomics applications. A defining difference between KLIC and COCA is that, while COCA performs a combination of the clusters found in each dataset, KLIC uses the similarities between data points observed in each dataset to perform the integrative step. Moreover, KLIC weights each dataset individually, which allows more informative datasets to be up-weighted relative to less informative ones, as demonstrated in our simulation study. Finally, we have used KLIC to integrate multiple 'omic datasets, in two different real world applications, finding biologically meaningful clusters. The results compare favourably to those obtained with COCA.

3.8.2 *Challenges*

The challenges typically encountered in clustering analysis concern the choice of the number of clusters and evaluating clustering results.

Choice of the number of clusters

Every clustering algorithm (except BHC) used in this chapter to produce consensus matrices requires the user to input a fixed number of clusters a priori. This can be problematic when the number of clusters is not known and the metrics commonly used to select K give ambiguous indications. In the next chapter we introduce a way of deriving kernel matrices from Bayesian mixture models which,

on top of being more sophisticated than the heuristic approaches used here, are able to take into account the uncertainty around the number of clusters.

Evaluating clustering results

Another recurring issue in cluster analysis is the evaluation of the clustering results. In simulation settings, when the true partition of the data is known, we were able to compare our clustering to the true one via the ARI. Moreover, external information was available for the transcriptional module discovery, which allowed us to make sure that our clusters are biologically meaningful. In many real applications, however, external information is not easily accessible. This is the case of the pan-cancer example, where expert knowledge would be required to validate the clinical importance of the cancer subtypes obtained using KLIC.

4

SUMMARISING AND COMBINING POSTERIOR SIMILARITY MATRICES DERIVED FROM MULTIPLE 'OMIC DATASETS

Here we propose a new method to summarise the posterior similarity matrices (PSMs) derived from the Markov chain Monte Carlo (MCMC) output of Bayesian model-based clustering. The idea is to see the posterior similarity matrix as a kernel matrix. Consequently, we are able to use kernel methods such as the kernel k-means algorithm presented in Chapter 3 to find a summary clustering of any posterior similarity matrix. Moreover, if we are in a setting where, as in the previous chapter, we have multiple different types of data for the same type of observations, we can initially perform Bayesian model-based clustering analyses on each dataset independently, and then combine and summarise all the posterior similarity matrices to obtain a global clustering of the data with the multiple kernel k-means algorithms presented in Chapter 3. We additionally show how we may include a response variable in order to perform *outcome-guided* (OG) integrative analyses, using SVMs. Both the unsupervised and the outcome-guided algorithms assign a weight to each dataset, that is output together with the global cluster assignment.

This work therefore contributes to the many ways of summarising the clusterings sampled from the posterior distribution that have already been proposed (Fritsch and Ickstadt, 2009; Wade and Ghahramani, 2018). Our approach performs equally well in the case of one dataset and has the advantage of being easily extended to the case of multiple data sources.

From a different perspective, this work also suggests a new, rational way to define kernels for performing clustering tasks of 'omics datasets. In Chapter 3, we generated kernels via consensus clustering. However, it has been shown by Şenbabaoğlu, Michailidis, and Li (2014) that CC has several limitations and should be used with caution. To give an instance, in one of the experiments performed in Şenbabaoğlu, Michailidis, and Li (2014), data were generated from a unimodal distribution, but the consensus matrix obtained via CC showed a clear clustering structure. Moreover, Lanckriet et al. (2004a) applied MKL methods to the problem of genomic data fusion, trying different kernels for each data source. We

showed in Chapter 3 that the choice of the kernel parameters is crucial and, to the best of our knowledge, there is currently no general strategy for choosing kernels that are optimal for clustering analyses. Therefore, how to define good "clustering kernels" in general remains an open problem. Using PSMs as kernels ensures that the similarities between data points reflect our model the clustering structure inferred using Bayesian model-based clustering.

We apply the methods developed here to the same real datasets considered in the previous chapter: multiplaform tumour subtyping and transcriptional module discovery. These two applications help us show that the methodologies presented in this chapter can lead to finding more refined tumour subtypes and more meaningful partitions of yeast genes.

Chapter outline

The chapter is organised as follows. In Section 4.1 we introduce the problem of summarising PSMs and prove that they are valid kernel matrices. In Section 4.2 we introduce the concept of outcome-guided integration of multi-omic datasets and explain how kernels methods can be used to combine multiple PSMs, both in the unsupervised and outcome-guided framework. In Section 4.3 we present some simulated examples of integration of posterior similarity matrices. In Sections 4.4 and 4.5 we show how our methodology can be applied to the datasets presented in Chapter 3. Section 4.6 summarises the main findings and challenges of this chapter.

4.1 SUMMARISING POSTERIOR SIMILARITY MATRICES

In this section, we briefly recall the concept of Bayesian mixture modelling and introduce the problem of summarising the posterior distribution on the cluster allocation. Then, we explain how the output of the MCMC algorithms for Bayesian mixture models can be used to obtain valid kernel matrices. This allows us to:

- use the kernel k-means algorithm to summarise a PSM and find a summary clustering of the data;
- combine multiple PSMs to perform integrative clustering of multiple datasets using the multiple kernel k-means algorithms presented in Chapter 3;
- use an external response variable to determine the weight of each dataset in the integrative setting, by using predictive kernel methods such as SVMs.

The SVM-based algorithms used for the last point are explained in detail in Section 4.2.

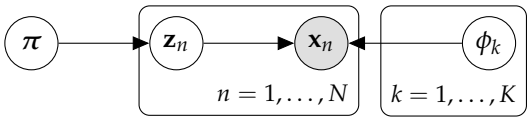


FIGURE 4.1: Finite mixture model.

In order to understand what PSMs are and why the question of how to summarise PSMs arises, we give a brief introduction to Bayesian mixture models and MCMC schemes. We then show that PSMs are valid kernel matrices.

4.1.1 *Bayesian mixture models*

Statistical methods for clustering can be divided into two main categories: heuristic approaches and model-based techniques. The former include the clustering algorithms used in the previous chapter, such as k-means (Hartigan and Wong, 1979) and hierarchical clustering (Kaufman and Rousseeuw, 1990). The latter are the focus of this chapter and are known as *mixture models* (McLachlan and Peel, 2004; Fraley and Raftery, 2002).

These models assume that the data are drawn from a mixture of distributions:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f_X(\boldsymbol{x}|\phi_k). \tag{4.1}$$

where f_X is a parametric density that depends on the parameter(s) ϕ_k and π_k are mixture weights such that $\sum_k \pi_k = 1$. These models are often fitted via the *expectation-maximisation* (EM) algorithm (Dempster, Laird, and Rubin, 1977), which gives as output point estimates for the parameters of each mixture component as well as cluster membership probabilities for each observation.

In the Bayesian framework, we assign a prior distribution to the set of all parameters $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k]$ and $\boldsymbol{\phi} = [\phi_1, \ldots, \phi_k]$ of Equation (4.1). This allows us to take into account the uncertainty around the values of the model parameters (see e.g. Rogers and Girolami, 2016, Chapter 10). In mathematical terms, a Bayesian mixture model with K components is defined as

$$\begin{aligned} x_n|z_n, \boldsymbol{\phi} &\sim F_X\left(\boldsymbol{x}_n|\phi_{z_n}\right) \\ z_n|\boldsymbol{\pi} &\sim \text{Categorical}\left(\boldsymbol{\pi}\right) \\ \boldsymbol{\phi}_k|H &\sim H \\ \boldsymbol{\pi}|\alpha &\sim \text{Dirichlet}\left(\frac{\alpha}{K}, \ldots, \frac{\alpha}{K}\right) \end{aligned} \tag{4.2}$$

where H is the prior distribution for the mixture parameters, and F_X is the probability distribution corresponding to the parametric density f_X, and $z_n \in \{1, \ldots, K\}$ are the cluster assignments of the data points $\boldsymbol{x}_n$, with $n = 1, \ldots, N$.

The posterior distributions of these models are usually not available in closed form. Inference on their parameters can be performed either via deterministic approximate inference methods, or using MCMC schemes (Gelfand and Smith, 1990). Deterministic approximate inference methods include the BHC algorithm introduced in Chapter 3 (Heller and Ghahramani, 2005), variational Bayes approaches (Bishop, 2006; Blei and Jordan, 2006), the sequential updating and greedy search (SUGS) algorithm of Wang and Dunson (2011), among others (for a more complete list of methods see Crook, Gatto, and Kirk, 2019). Here we restrict our attention to MCMC schemes, which are used to construct a Markov chain that has as the posterior distribution of the model given the data as its invariant density. Once the chain has reached convergence, the MCMC are effectively samples from the posterior distribution of our model.

Another advantage of adopting the Bayesian viewpoint when dealing with mixture models is that we can seek to infer the number of mixture components, rather than having to specify this a priori. One way of doing this is to make use of reversible jump MCMC (Green, 1995), which allows us to sample from parameter spaces of varying dimensions and can therefore be used to jump between mixtures with different numbers of components (Richardson and Green, 1997). Another approach is to choose a number of components K that is large compared to the number of observations. Not all the components of the mixtures need to be occupied, therefore K only places an upper bound on the number of clusters. Rousseau and Mengersen (2011) showed that in these so-called *overfitted mixture models*, in the posterior distribution the components in excess are left empty (provided that a "reasonable" prior is specified). Alternatively, one can consider the limit $K \to \infty$ in Equation (4.2), giving rise to a *Dirichlet process mixture model* (DPMM; Rasmussen, 2000; Neal, 2000) that is a mixture model having a Dirichlet process (DP) as the prior on the mixture components.

We recall here the definition of DP (Ferguson, 1973):

Definition 4.1 *Let $(\Omega, \mathcal{B})$ be a measurable space with state space Ω and σ-field $\mathcal{B}$ and let H be a probability measure on $(\Omega, \mathcal{B})$. A* Dirichlet process *is a random probability measure G on $(\Omega, \mathcal{B})$ such that for any finite partition $(T_1, \ldots, T_F)$ of Ω, it holds*

$$(G(T_1), \ldots, G(T_F)) \sim \textit{Dirichlet}\,(\alpha H(T_1), \ldots, \alpha H(T_F))\,.$$

We indicate this by

$$G \sim \mathrm{DP}\,(\alpha, H)\,,$$

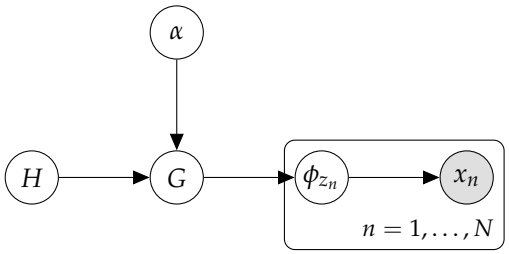


FIGURE 4.2: Dirichlet process mixture model.

where $\alpha \in \mathbb{R}^+$ is called the *concentration parameter* and H is the *base measure*. The resulting mixture model can be written as

$$\begin{aligned} x_n|\phi_n &\sim F_X(x_n|\phi_n) \\ \phi_n|G &\sim G \\ G|\alpha, H &\sim \text{DP}(\alpha, H) \end{aligned}$$

where ϕ_n, $n = 1, \ldots, N$ are the mixture parameters for observation x_n, with $n = 1, \ldots, N$. Observations that have the same value of ϕ_n belong to the same cluster.

Constructive representations of the Dirichlet prior include the stick-breaking construction of Sethuraman (1994) and the Chinese restaurant process, which has been attributed to Jim Pitman (see Aldous, 1985). In this work, we use the R package `PReMiuM` of Liverani et al. (2015) where the stick breaking construction is used, the `DPMSysBio` Matlab[1] toolbox of Žurauskienė, Kirk, and Stumpf (2016) which exploits the Chinese restaurant process. We also use MDI (Kirk et al., 2012), which makes used of the overfitted mixtures of Rousseau and Mengersen, 2011. Throughout this thesis, we employ the C implementation of MDI, rather than the original Matlab package, because it is more efficient and therefore more suitable for the large datasets considered in this thesis (Mason et al., 2016).

Before proceeding with the application of DPMMs to real data problems, it is important to note that, while these models do not require the specification of the number of clusters K, they do involve a parameter α that influences the prior expectation of the number of clusters. Moreover, the *rich-get-richer* property of these models means that, a priori, users expect to see a small number of large clusters (Vlachos, Korhonen, and Ghahramani, 2009). Other known issues of DPMMs are that MCMC schemes are used to fit these models, which can be computationally costly and may get stuck in local modes (Jain and Neal, 2004).

Profile regression

Later in this chapter, we also use an extension of DPMMs, known as *profile regression* (Molitor et al., 2010; Papathomas et al., 2011; Liverani et al., 2015). The idea

[1] `https://uk.mathworks.com/products/matlab.html`

of profile regression is that, if a response y_n is available for each $n = 1, \ldots, N$, the observations $d_n = (x_n, y_n)$ are jointly modelled as the product of the response model and a covariate model. The resulting likelihood is

$$p(d_n|\phi_n) = f_Y(y_n|\phi_n) f_X(x_n|\phi_n),$$

where f_Y is a parametric density and ϕ_n includes the parameters of both f_X and f_Y (Figure 4.3). This can be useful in situations where a response variable that is believed to be associated to the clustering structure of interest is available. In fact, the inclusion of this response into the model allows us to influence the clustering, favouring the partitions of the data that group together data points with similar response values.

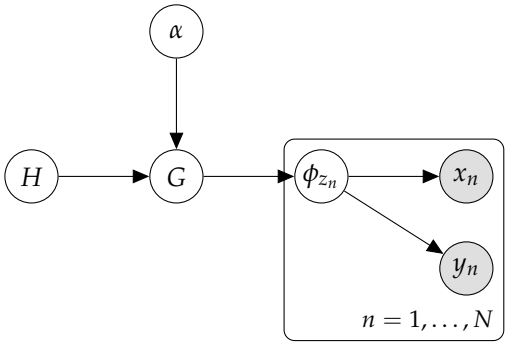


FIGURE 4.3: Profile regression.

4.1.2 *Posterior similarity matrices*

When using MCMC methods in order to perform Bayesian clustering on a dataset $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, one obtains a vector of cluster assignments $\boldsymbol{c}^{(b)} = [c_1^{(b)}, \ldots, c_N^{(b)}]$ from the posterior distribution for each iteration of the algorithm $b = 1, \ldots, B$ (see, for example, Neal, 2000). From this, it is possible to obtain a Monte Carlo estimate of the probability that observations i and j belong to the same cluster as follows:

$$P(c_i = c_j|X) \approx \frac{1}{B} \sum_{b=1}^{B} \mathbb{1}\left(c_i^{(b)} = c_j^{(b)}\right) =: \Delta_{ij}, \tag{4.3}$$

where $\mathbb{1}$ is the indicator function. We denote by Δ the PSM that is the matrix that has ijth entry Δ_{ij} equal to the right hand side of Equation (4.3).

Many ways to find a final clustering using the using the MCMC cluster allocation samples have been proposed (Binder, 1978; Dahl, 2006; Fritsch and Ickstadt, 2009; Medvedovic and Sivaganesan, 2002; Wade and Ghahramani, 2018). A simple solution is to choose, among the $\boldsymbol{c}^{(b)}$, the one that maximises the posterior density. The problem with this approach is that many clusterings are associated with very

similar posterior densities (Fritsch and Ickstadt, 2009). A more principled approach is to define a loss function $L(\boldsymbol{c}, \hat{\boldsymbol{c}})$ measuring the loss of information that occurs when estimating the true clustering $\boldsymbol{c}$ with $\hat{\boldsymbol{c}}$ (Binder, 1978). The optimal clustering $\boldsymbol{c}^*$ is then defined as the one minimising the posterior expected loss:

$$\boldsymbol{c}^* = \arg\min_{\hat{\boldsymbol{c}}} \mathbb{E}\left[L(\boldsymbol{c}, \hat{\boldsymbol{c}})|X\right] = \arg\min_{\hat{\boldsymbol{c}}} \sum_{\boldsymbol{c}} L(\boldsymbol{c}, \hat{\boldsymbol{c}}) p\left(\boldsymbol{c}|X\right).$$

Binder (1978), for instance, suggested choosing the clustering $\hat{\boldsymbol{c}}$ that minimises the loss function

$$L_{\text{Binder}}(\boldsymbol{c}, \hat{\boldsymbol{c}}) = \sum_{i<j} \left[l_1 \mathbb{1}(c_i = c_j)\mathbb{1}(\hat{c}_i \neq \hat{c}_j) + l_2 \mathbb{1}(c_i \neq c_j)\mathbb{1}(\hat{c}_i = \hat{c}_j)\right],$$

where l_1 and l_2 are positive constants determining whether assigning observations that belong to the same clusters to different clusters is penalised more highly than assigning observations that belong to different clusters to the same cluster ($l_1/l_2 > 1$) or vice versa ($l_1/l_2 < 1$). If $l_1 = l_2$, then

$$\boldsymbol{c}^*_{\text{Binder}} = \arg\min_{\hat{\boldsymbol{c}}} \sum_{i<j} |\mathbb{1}\left(\hat{c}_i = \hat{c}_j\right) - \Delta_{ij}|.$$

More recently, Wade and Ghahramani (2018) proposed an alternative to Binder's loss function based on the *variation of information* of Meilă (2007):

$$L_{\text{VI}}(\boldsymbol{c}, \hat{\boldsymbol{c}}) = H(\boldsymbol{c}) + H(\hat{\boldsymbol{c}}) - 2I(\boldsymbol{c}, \hat{\boldsymbol{c}}),$$

where $H(\boldsymbol{c})$ and $H(\hat{\boldsymbol{c}})$ represent the entropy of clusterings $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ respectively, $I(\boldsymbol{c}, \hat{\boldsymbol{c}})$ is the mutual information between clusterings $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$. Defining by C_i the set of elements that belong to cluster i in clustering $\boldsymbol{c}$, $\hat{C}_j$ the set of elements that belong to cluster j in clustering $\hat{\boldsymbol{c}}$, $n_{ij} = |C_i \cap \hat{C}_j|$, $n_{i+} = \sum_j n_{ij}$, $n_{+j} = \sum_i n_{ij}$, this is

$$L_{\text{VI}} = -\sum_{i=1}^{K_N} \frac{n_{i+}}{N} \log_2\left(\frac{n_{i+}}{N}\right) - \sum_{j=1}^{\hat{K}_N} \frac{n_{+j}}{N} \log_2\left(\frac{n_{+j}}{N}\right) - 2\sum_{i=1}^{K_N}\sum_{j=1}^{\hat{K}_N} \frac{n_{ij}}{N} \log_2\left(\frac{n_{ij}N}{n_{i+}n_{+j}}\right),$$

where K_N and $\hat{K}_N$ represent the number of clusters in $\boldsymbol{c}$ and $\hat{\boldsymbol{c}}$ respectively.

Dahl (2006) advanced the idea to choose, among all the clustering vectors $\boldsymbol{c}^{(b)}$, the one that minimises the least-squared distance to the PSM:

$$\boldsymbol{c}^*_{\text{Dahl}} = \arg\min_{\hat{c}} \sum_{i<j} \left[\mathbb{1}\left(\hat{c}_i = \hat{c}_j\right) - C_{ij}\right]^2.$$

This turned out to be equivalent to minimising Binder's loss function (Equation 4.1.2). A similar approach was developed by Wang and Porter (2018) using non-negative matrix factorisation. Fritsch and Ickstadt (2009) improved on the methods of Binder and Dahl by maximising the posterior expected ARI of Hubert and Arabie (1985).

Moreover, Medvedovic and Sivaganesan (2002) applied the complete linkage approach of Everitt (1993) to the matrix of pseudo-distances $1 - \Delta$, while Molitor et al. (2010) used the PAM algorithm (Kaufman and Rousseeuw, 1990).

All these methods are only applicable with one PSM. In what follows we describe a new way to find a clustering using PSMs that also allows us to summarise multiple similarity matrices Δ_m and find a global clustering.

4.1.3 *Identifying posterior similarity matrices as kernel matrices*

Posterior similarity matrices are computed as the element-by-element average of the B co-clustering matrices $C^{(b)}$, defined as:

$$C_{ij}^{(b)} = \begin{cases} 1 & \text{if } c_i^{(b)} = c_j^{(b)}, \\ 0 & \textit{otherwise}, \end{cases}$$

derived from each iteration of the MCMC chain: $C_{ij}^{(b)}$ indicates whether the statistical units i and j are assigned to the same cluster at iteration b. Therefore, PSMs are co-clustering matrices like those considered in Chapter 3 and the argument that we used in Section 3.4 to prove that any co-clustering matrix is a valid kernel holds for PSMs too.

4.2 COMBINING POSTERIOR SIMILARITY MATRICES

Section 4.2.1 deals with the unsupervised integration of multiple PSMs derived from different 'omic layers. In Section 4.2.2 is introduced the concept of outcome-guided integration.

4.2.1 *Unsupervised integration*

Having shown that PSMs are valid kernels, it follows that unsupervised integration can be performed using KLIC, in the same way as in Chapter 3 for consensus matrices. This can be useful for many reasons. First, MCMC schemes are computationally intensive: while it is possible in principle to define mixtures that accommodate multiple 'omic datasets, it is often infeasible to run MCMC chains for the full multi-omic datasets. Being able to perform inference on each 'omic layer separately and to combine the PSM at a later time requires less time and computational power. On top of that, as is the case for consensus matrices, assigning different weights to each kernel allows us to assess how much each dataset contributed to the final clustering. This can give an idea of how much information is present in each data type about the clustering structure.

4.2.2 *Outcome-guided integration*

The problem with combining multiple kernels is that, as we have seen in Chapter 3, it is not always clear whether they all define the same clustering structure. To overcome this issue, we also propose an outcome-guided algorithm to summarise multiple PSMs. The idea is that, instead of choosing the weight of each kernel in an unsupervised way, if we have a variable available which is closely related to the outcome of interest, we should weight more highly the kernels in which statistical units that have similar outcomes are closer to each other. This allows us to uncover the clustering structure in the data that is most similar to the one defined by the response variable, in a similar spirit to that of profile regression.

Suppose that, in addition to the PSMs $\Delta_1, \ldots, \Delta_M$, we also have a categorical response variable y_n associated with each observation $\boldsymbol{x}_n$. As we explained above, we would like to use this information to guide our clustering algorithm. We can use the *SimpleMKL* algorithm described in the remainder of this section to find the kernel weights $\theta_1, \ldots, \theta_m$ and then use kernel k-means on the weighted kernel $\Delta = \sum_{m=1}^{M} \theta_m \Delta_m$ to find the final clustering (Figure 4.4).

We refer to the approach presented here as *outcome-guided KLIC*, as opposed to the *unsupervised* version of KLIC introduced in Chapter 3. Approaches of this type are also referred to as *semi-supervised* (Bair and Tibshirani, 2004; Koestler et al., 2010). However, we avoid using this term here, since it is more commonly used to indicate machine learning approaches that combine large amounts of unlabelled data with a few labelled data points (Yu et al., 2006).

Related methods

The importance of exploiting clinical information when performing clustering of 'omic data has been highlighted by multiple publications (see e.g. Ahmad and Fröhlich, 2017; Chaudhary et al., 2018). However, to the best of our knowledge, the only approach that takes a comparable approach to ours is the outcome-weighted integrative clustering method of Arora et al. (2020), called *survClust*, which was developed to identify cancer subtypes using multi-omic and survival data. In *survClust* the distance between two statistical units $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^P$ is defined as

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left[\left(\boldsymbol{x}_i - \boldsymbol{x}_j \right)^T W \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right]^{1/2},$$

where W is a $P \times P$ diagonal weight matrix. The diagonal entries of of the weight matrix are indicated by w_p, $p = 1, \ldots, P$ and are computed as the logarithm of the absolute value of the hazard ratio, obtained by fitting a univariate Cox proportional hazards model (Cox, 1972) for each feature p. As a result, $w_p = 0$ if feature p is not associated with survival and w_p is large if there is evidence that feature p is associated with survival. Once these weighted distances are

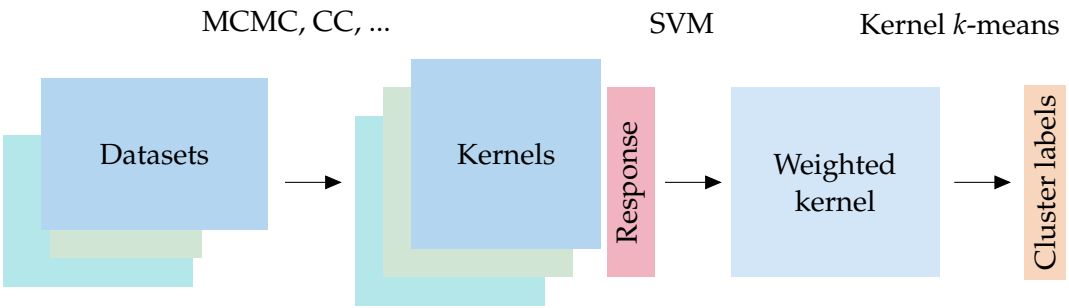


FIGURE 4.4: Schematic representation of outcome-guided KLIC. Each colour indicates a different dataset/kernel. First, a mixture model is fit on each dataset separately. The resulting PSMs are valid kernels that can be used as input to *simpleMKL*, if a response variable is available, to determine the kernel weights. The weighted kernel is then used as input to kernel k-means to find the final clustering of the data. The same can be done with any other type of kernels, including those generated via CC.

calculated, a clustering is obtained by projecting the data into a lower dimensional space and clustering the projected data points via the k-means algorithm. There are several differences between *survClust* and the outcome-guided version of KLIC. First, *survClust* can only be used with survival data, while our approach assumes that the response variable is categorical (and extensions to continuous responses should be straightforward, as we explain later in this section). Secondly, outcome-guided KLIC assigns different weights to each 'omic layer, while *survClust*'s weights are feature specific.

The remainder of this section is dedicated to introducing SVMs and explaining how they can be used to extend KLIC to perform outcome-guided integration.

Support vector machines

We briefly recall here the concept of SVM (Boser, Guyon, and Vapnik, 1992) that is widely used for solving problems in classification and regression (Schölkopf and Smola, 2001; Bishop, 2006).

In its simplest form, this method is applied to a binary classification problem, in which the data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^P$ in the training set are assigned to two classes indicated by the target values $y_n \in \{-1, 1\}, n = 1, \ldots, N$. We consider a feature map $\phi : \mathbb{R}^P \to \mathcal{X}$ and the associated symmetric PSD kernel $\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that $\delta(\boldsymbol{x}, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle_{\mathcal{X}}$. Suppose that there exist some values of α_n and b such that

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} \alpha_n \delta(\boldsymbol{x}, \boldsymbol{x}_n) \tag{4.4}$$

satisfies $f(\boldsymbol{x}_n) + b > 0$ if $y_n = 1$ and $f(\boldsymbol{x}_n) + b < 0$ otherwise. f is a function that lives in a function space $\mathcal{H}$ endowed with the norm $\|\cdot\|_{\mathcal{H}}$. Then, this function can be used to classify new data points $\boldsymbol{x}$ according to the sign of $f(\boldsymbol{x}) + b$.

For support vector machines, the parameters α_n and b are chosen so as to maximise the *margin*, i.e. the distance between the decision boundary given by Equation (4.4) and the point $\boldsymbol{x}_n$ that is closest to the boundary (Figure 4.5) . It can be shown that this can be achieved by solving the QP problem (see e.g. Bishop, 2006; Rakotomamonjy et al., 2008)

$$\begin{aligned} \underset{f,\, b}{\text{minimise}} \quad & \frac{1}{2}\|f\|_{\mathcal{H}}^2 \\ \text{subject to} \quad & y_n\big[f(\boldsymbol{x}_n) + b\big] \geq 1, \quad \forall n. \end{aligned} \tag{4.5a}$$

However, in real applications, it is usually not possible to separate the two classes perfectly. Hence, in order to take into account misclassifications, it is necessary to introduce a penalty term that is linear with respect to the distance of the misclassified points to the classification boundary (Bennett and Mangasarian, 1992). To this end, we define a variable ξ_n (known as a *slack variable*) for each data point such that

$$\xi_n = \begin{cases} 0, & \text{if } \boldsymbol{x}_n \text{ is correctly classified,} \\ |y_n - f(\boldsymbol{x}_n)|, & \text{otherwise.} \end{cases}$$

The optimisation problem of Equation (4.5) then becomes

$$\begin{aligned} \underset{f,\, b,\, \{\xi_n\}}{\text{minimise}} \quad & \frac{1}{2}\|f\|_{\mathcal{H}}^2 + C\sum_n \xi_n \\ \text{subject to} \quad & y_n\big[f(\boldsymbol{x}_n) + b\big] \geq 1 - \xi_n, \quad \forall n, \\ & \xi_n \geq 0, \quad \forall n, \end{aligned} \tag{4.6a}$$

where $C > 0$ is a parameter that controls the penalisation of misclassifications. The objective functions (4.5a) and (4.6a) are quadratic, so any local optimum is also a global optimum. One of the most popular approaches to solve this type of problems is *sequential minimal optimisation* (Platt, 1999). For more details about SVMs see, for instance, Bishop (2006).

Multiple kernel learning for support vector machines

In the multiple kernel learning framework for SVMs, we consider again M different feature representations, with mapping functions ϕ_m and corresponding kernel functions δ_m and feature spaces $\mathcal{X}_m$. We replace the kernel δ of Equation (4.4) with a convex combination of kernels δ_m (Lanckriet et al., 2004a):

$$f(\boldsymbol{x}) + b = \sum_{m=1}^{M} \theta_m f_m(\boldsymbol{x}) + b$$

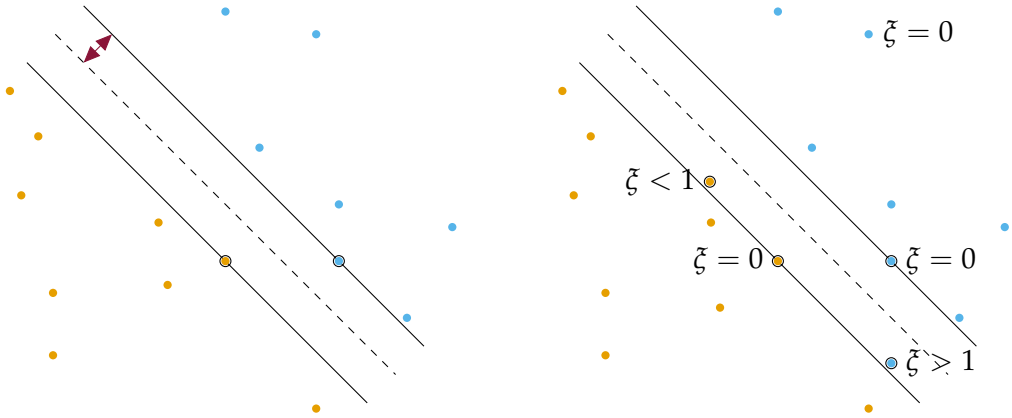


FIGURE 4.5: Illustration of a support vector machine. Data points are represented in the feature space $\mathcal{X}$, each colour indicates one of the two classes. The decision boundary is represented by the dashed black line. Left: the two classes are separable. The red double arrow indicates the margin, which is the distance between the decision boundary and the data point that is closer to the boundary. Right: the classes are not separable. $\xi = 0$ for all data points that are correctly classified, $0 < \xi < 1$ for those that lie within the margin but on the correct side of the boundary, $\xi > 1$ for the points that are on the incorrect side of the boundary. Figure freely adapted from Bishop (2006, Chapter 7).

where $\theta_m \geq 0$, $\sum_m \theta_m = 1$ and $f_m = \sum_n \delta_m(\boldsymbol{x}, \boldsymbol{x}_n)$. Rakotomamonjy and Bach (2007) proposed then to solve the optimisation problem

$$\begin{aligned}
\underset{\{f_m\}, b, \{\xi_n\}, \{\theta_m\}}{\text{minimise}} \quad & J(\boldsymbol{\theta}) := \frac{1}{2} \sum_{m=1}^{M} \frac{1}{\theta_m} \|f_m\|^2_{\mathcal{H}_m} + C \sum_n \xi_n && \text{(4.7a)} \\
\text{subject to} \quad & y_n \left[\sum_{m=1}^{M} f_m(\boldsymbol{x}_n) + b \right] \geq 1 - \xi_n, \quad \forall n, \\
& \xi_n \geq 0, \quad \forall n, \\
& \sum_m \theta_m = 1, \\
& \theta_m \geq 0, \quad \forall m
\end{aligned}$$

using the convention that $x/0 = 0$ if $x = 0$ and ∞ otherwise. The algorithm of Rakotomamonjy and Bach takes the name of *simpleMKL* and is based on the idea that one can iteratively solve a standard SVM problem (4.6a) for a fixed value of $\boldsymbol{\theta}$ and then update the vector of weights $\boldsymbol{\theta}$ using the gradient descent method on the objective function $J(\boldsymbol{\theta})$. Since the objective function is smooth and differentiable with Lipschitz gradient, it can be easily optimised with the reduced gradient algorithm (Luenberger and Ye, 1984, Chapter 11). If the standard SVM problem is

solved exactly at each iteration, then convergence to the global optimum is guaranteed (Luenberger and Ye, 1984).

Multi-class multiple kernel learning

SVMs can be used also when the target value y_n takes more than two different values. The most commonly used approaches are called *one-versus-one* (Knerr, Personnaz, and Dreyfus, 1990) and *one-versus-the-rest* (Vapnik, 1999). In the first one, we consider in turn each class as the "positive" case (with target value $+1$), and all the others as the "negative" cases (with target value -1). This way, if y_n takes Q distinct values, we construct Q different classifiers and then assign a new observation $\boldsymbol{x}$ using

$$y(\boldsymbol{x}) = \max_{q \in \{1,\dots,Q\}} y_k(\boldsymbol{x}).$$

The second approach is to train one SVM for each pair of classes and then assign a point $\boldsymbol{x}$ to the class to which it is assigned more often.

Rakotomamonjy et al. (2008) extended the *SimpleMKL* algorithm to the case of a response with $K > 2$ classes. Both these approaches can be used with the *SimpleMKL* algorithm, defining a new cost function $J(\boldsymbol{\theta})$ as the sum of all the cost functions of the partial SVMs $J_s(\boldsymbol{\theta})$:

$$J(\boldsymbol{\theta}) = \sum_{s \in \mathcal{S}} J_s(\boldsymbol{\theta}),$$

where $\mathcal{S}$ indicates the set of all partial SVMs and each J_s is defined as in Equation (4.7a).

Just like the methodology introduced in Chapter 3, none of the approaches presented here explicitly rely on the fact that the Δ_m are PSMs or consensus matrices. Hence, any other type of matrix can be used, as long as it is symmetric, positive semi-definite and the entries Δ_{ij}^m can be interpreted as some measure of the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

4.3 SIMULATION STUDY

Here we show how the methods presented above perform in practice. We generate four synthetic datasets, each composed of data belonging to six different clusters of equal size. Each observation $\boldsymbol{x}_n^{(k)} \in \{0, 1, 2\}^{10}$ belonging to cluster k is drawn from a multivariate categorical distribution such that, for each covariate $j = 1, \dots, 10$,

$$x_{nj}^{(k)} \sim \text{Categorical}(\pi_{1k}, \pi_{2k}, \pi_{3k}),$$

where π_{ik}, $i = 1, 2, 3$ are such that $\pi_{ik} = w\rho_{ik} + (1 - w)/3$, with $[\rho_{1k}, \rho_{2k}, \rho_{3k}] \sim$ Dirichlet(0.01). Each dataset has a different value of $w \in [0, 1]$. Higher values of w_d give clearer clustering structures. The response variable is binary, with $P(y_n = 1|z_n = k) = \theta_k$, where $\theta_k \in \{0.01, 0.1, 0.15, 0.85, 0.9, 0.99\}$. We repeat each experiment 100 times. For each synthetic dataset, we use the MCMC algorithm for Dirichlet process mixture models implemented in the R package `PreMiuM` of Liverani et al. (2015) to obtain the PSMs. We use unsupervised discrete mixtures (Liverani et al., 2015, Section 3.2) except in one setting (detailed below) where the profile regression model of Molitor et al. (2010) is employed, using a discrete mixture with categorical response. In both cases we use the default hyperparameters, which we found to work well in practice.

We consider four different simulation settings:

Setting A: same structure in every dataset. The clustering structure in every dataset is the same and is related to the outcome of interest. Each dataset has a different level of cluster separability, obtained setting $w = 0.2, 0.4, 0.6, 0.8$. As in Chapter 3, we refer to the dataset generated with $w = 0.2$, which has almost no cluster separability, as dataset 0, the one generated with $w = 0.4$, which has low cluster separability, as 1, and so on. One set of PSMs used for this setting is shown in Figure 4.6.

Setting B: same structure in every dataset with extra covariates. As in setting A, the clustering structure in each dataset is the same and each dataset has a different level of cluster separability. In this case, however, each dataset contains some additional covariates that have no clustering structure.

Setting C: one irrelevant dataset. The dataset with highest cluster separability has a clustering structure that is unrelated to the response variable, all the other datasets are the same as in setting A.

Setting D: profile regression. This is the same as setting C, but profile regression is used to derive the PSMs.

In Section 4.3.1 we show that the kernel k-means approach applied to a PSM derived from a single dataset performs similarly to standard clustering methods. Additionally, in Section 4.3.2 we apply the developed methods to the synthetic datasets in the unsupervised and outcome-guided framework.

4.3.1 *Summarising posterior similarity matrices*

Before using the PSMs with the kernel-based integrative clustering methods, we carry out some simulation studies to ensure that the kernel k-means algorithm

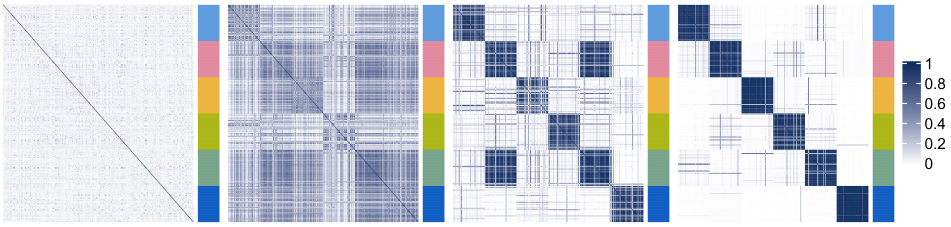


FIGURE 4.6: PSMs of the datasets used for setting A. The rows and columns correspond to the statistical units. The coloured bar on the right of each PSM represents the true clusters. The values of w used to generate these matrices are, from left to right, 0.2, 0.4, 0.6, and 0.8.

performs equally well at summarising the MCMC output as the other methods from the literature. The advantage is that the kernel k-means can be extended to combine multiple datasets.

We use the synthetic datasets described in setting A above and compare the kernel k-means algorithm to the methods implemented in the R package `mcclust` (Fritsch and Ickstadt, 2009). All these methods take a PSM Δ as input and find the clustering $\boldsymbol{c}^*$ that maximises the *posterior expected Rand index* (PEAR). This is achieved by choosing the clustering $\boldsymbol{c}^*$ that maximises the following quantity:

$$\frac{\sum_{i<j} \mathbb{I}_{\{c_i^*=c_j^*\}} \Delta_{ij} - \sum_{i<j} \mathbb{I}_{\{c_i^*=c_j^*\}} \sum_{i<j} \Delta_{ij} \big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_{i<j} \mathbb{I}_{\{c_i^*=c_j^*\}} + \sum_{i<j} \Delta_{ij}\right] - \sum_{i<j} \mathbb{I}_{\{c_i^*=c_j^*\}} \sum_{i<j} \Delta_{ij} \big/ \frac{n}{2}}. \tag{4.8}$$

The clusterings $\boldsymbol{c}^*$ taken into consideration by these methods can be chosen in different ways. We use hierarchical clustering with $1 - \Delta$ as distance matrix with average and complete linkage. The `maxpear` function tries all the possible numbers of clusters between one and a maximum number of clusters $K_{\max}$ specified by the user. We consider both 6 and 20 as values for $K_{\max}$. We also use the `maxpear` function to try with all the clusterings in the MCMC output and take the one that maximises the quantity of Equation (4.8). We repeat this procedure using the `minVI` function of the R package `mcclust.ext` of Wade and Ghahramani (2018), where the selected $\boldsymbol{c}^*$ is the one that minimises the lower bound for the posterior expected variation of information (Equation 4.1.2) from Jensen's inequality.

Figure 4.7 shows the box plots of the adjusted Rand index obtained repeating the experiment with 100 different sets of synthetic data calculated for each of the considered clustering algorithms and for all values of w. We can see that the kernel k-means on average performs better than the other methods when the number of clusters is known and that it performs similarly to the others when the number of clusters is chosen to maximise the average silhouette.

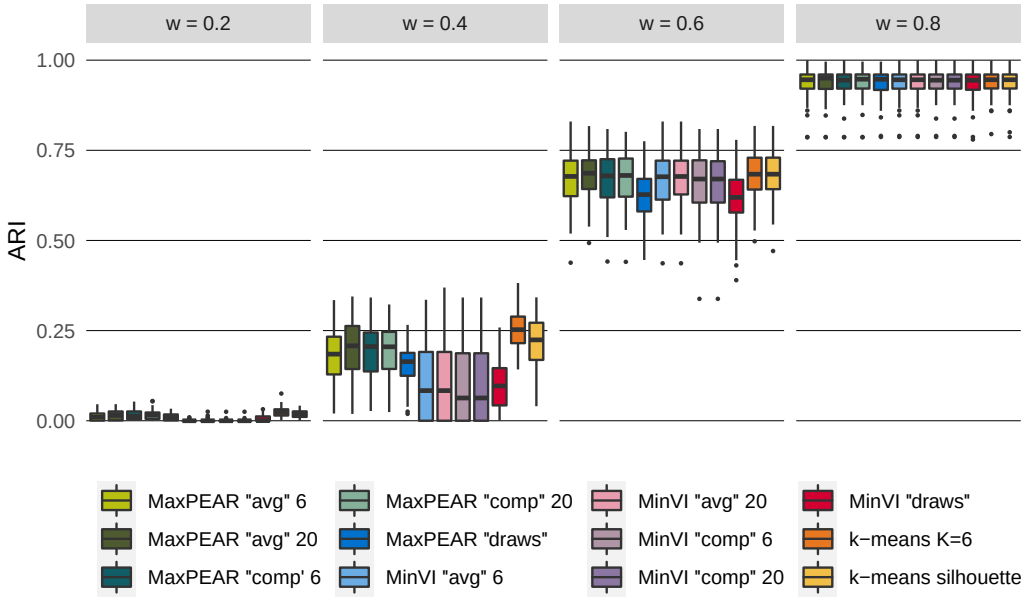


FIGURE 4.7: ARI for the kernel k-means applied to one dataset at a time, for different values of ρ compared to maximising the PEAR as suggested by Fritsch and Ickstadt (2009) and to minimising the VI as suggested by Wade and Ghahramani (2018). For both methods, we try different settings, namely: performing hierarchical clustering on the matrix $1 - \Delta$ with average ("avg") and complete ("comp") linkage, with maximum number of clusters equal to either 6 or 20, as well as considering all the clusterings samples that are appear in the MCMC output ("draws"). For kernel k-means, the results obtained fixing the number of clusters to six and choosing it via the silhouette are presented.

4.3.2 *Integrative clustering*

We assess the MKL-based unsupervised and outcome-guided integrative approaches in the four settings described above. For each setting we consider four different subsets of data, each combining three out of our four synthetic datasets. In what follows, we indicate by "0+1+2" the integration of the datasets generated with values of w equal to 0.2, 0.4, and 0.6 respectively, and similarly for the other combinations of datasets. Here we show the ARI between the clusterings found via MKL integration and the true cluster labels, the weights assigned to each dataset in each setting are instead reported in Appendix C.

Setting A: same structure in every dataset. The ARI obtained by combining the datasets in the unsupervised and outcome-guided frameworks is shown in the first row of Figure 4.8. The values of the ARI obtained in the previous section on each dataset separately are also reported. In all settings we set the number of clusters to the true value, six. The unsupervised integration performed using localised multiple kernel k-means allows to reach values of the ARI that are close to those of the "best" dataset (i.e. the dataset that has the highest value of cluster separability) among the three datasets in each subset. This is because the unsupervised MKL approach considered here assigns higher weights to the datasets that give rise to kernels with higher values of ρ, which in this case correspond to higher values of w. Moreover, even higher values of the ARI are achieved via outcome-guided integration, as a result of taking into account the outcome when weighting the datasets. In this case, the kernels that help separate the classes in the response have higher weights than the others.

Setting B: same structure in every dataset with extra covariates. In the second row of Figure 4.8 are shown the results obtained for setting B, where the PSMs are obtained exploiting (an adaptation of) the variable selection strategy of Chung and Dunson (2009) implemented in the R package `PReMiuM`. Despite the fact that the ARI of dataset 2 is lower than in the previous case, the integration results are better than in Setting A. Again, this is due to the fact that most informative kernels are weighted more highly than the other ones.

Setting C: one irrelevant dataset. This simulation study helps us to show that the outcome-guided approach favours the clustering structures that agree with the structure in the response. For this reason, we use a dataset with high cophenetic correlation coefficient whose clustering structure is not related to the response. The results are presented in the third row of Figure 4.8. Again, localised multiple kernel k-means assigns higher weights to the datasets that are more easily separable, i.e. datasets that give rise to kernels having higher cophenetic correlation coefficients. Note that here higher values of w correspond to higher cophenetic

correlation. In this situation, this causes the ARI of the subsets of kernels that include dataset 3 to drop to zero. In the outcome-guided case, instead, the dataset that has the highest level of cluster separability but is not related to the outcome of interest has (almost) always weight equal to zero.

Setting D: profile regression. Lastly, we consider the case where profile regression is used, instead of regular DPMMs (fourth row of Figure 4.8). We see that, as expected, the ARI is higher than in the previous cases for the clustering obtained with each dataset taken separately, except of course for dataset 3, that has a different clustering structure. This is reflected in an improvement of the ARI of the unsupervised and outcome-guided integration, for all considered subsets of data. In particular, the latter almost always allows to retrieve the true clustering.

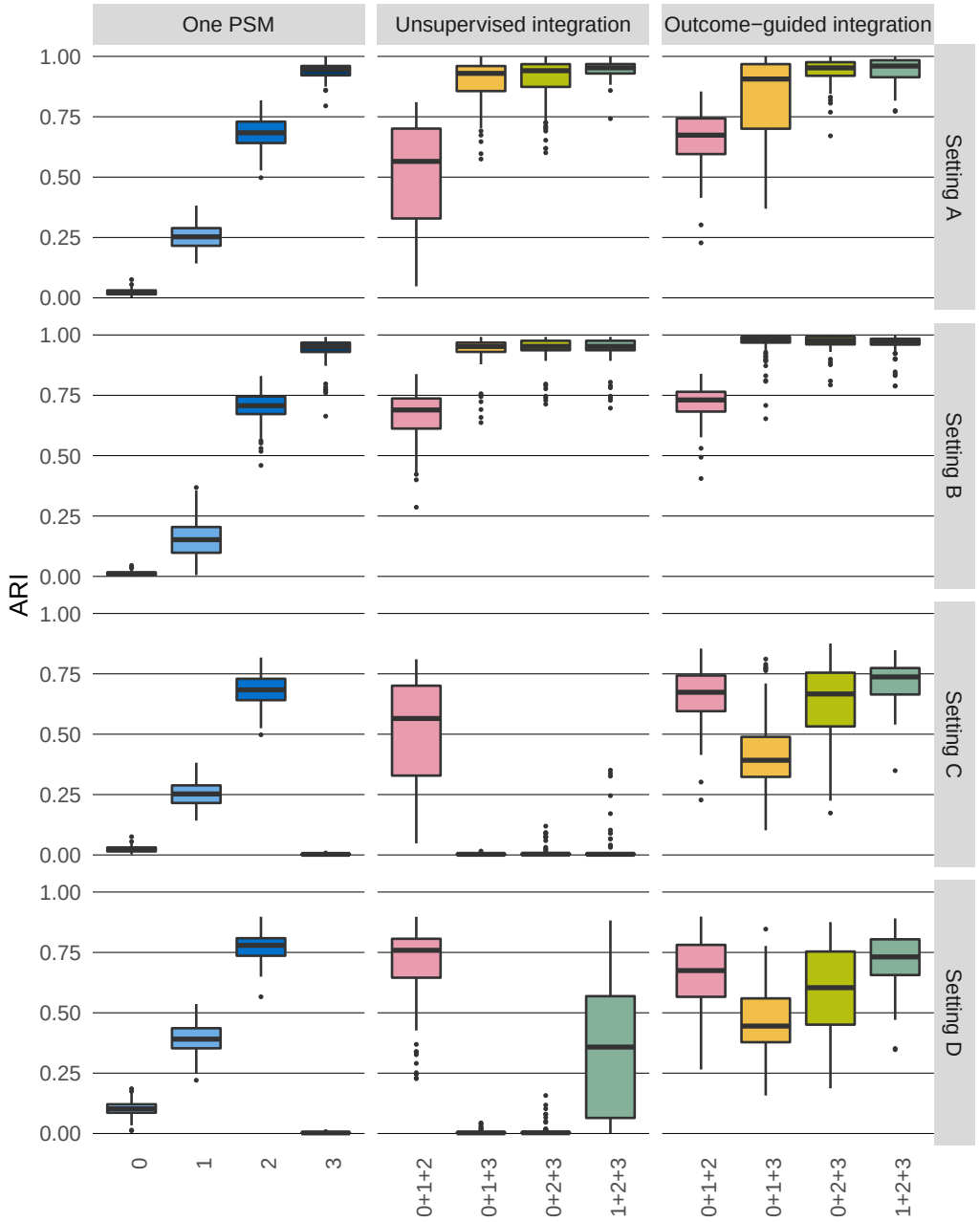


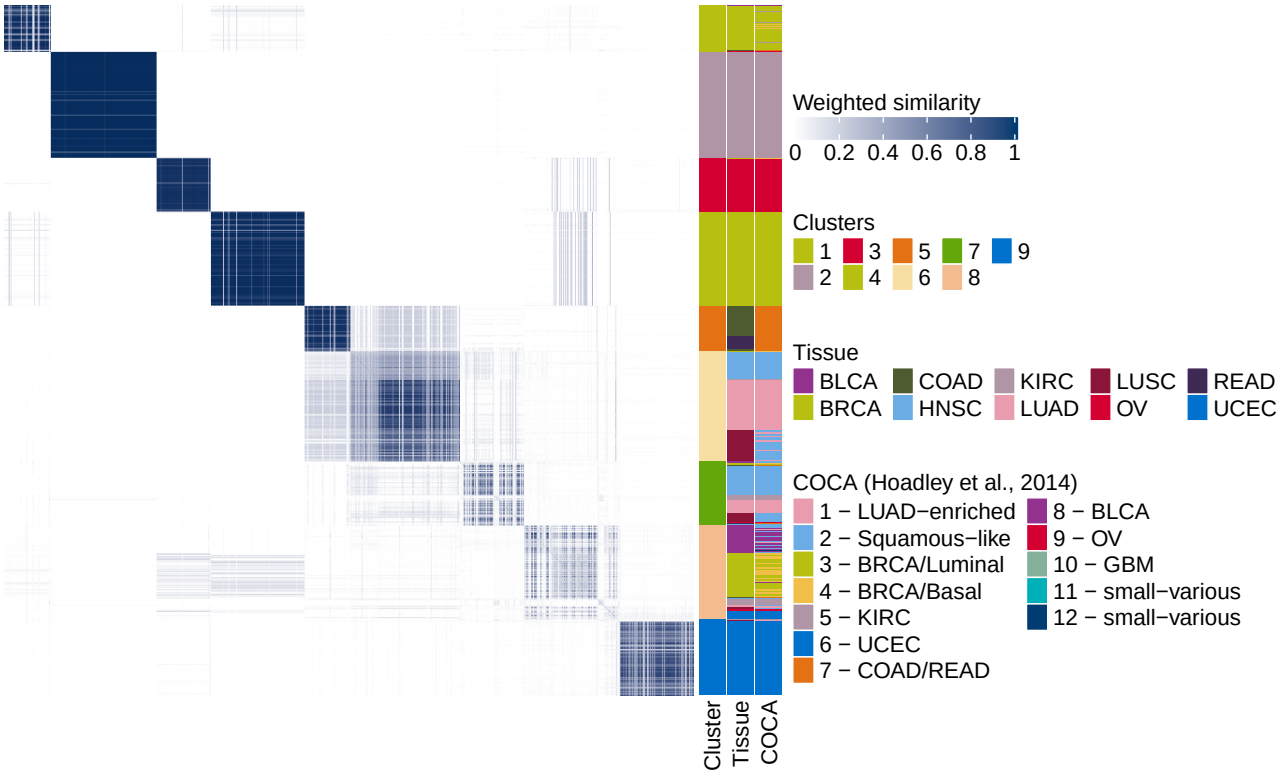
FIGURE 4.8: Simulation study, ARI obtained by summarising the PSMs one at a time using kernel k-means (left), combining different subsets of three PSMs in an unsupervised fashion using localised multiple kernel k-means (centre), and combining the same subsets making use of a response variable and multi-class SVMs to determine each PSM's weight and using kernel k-means for the final clustering (right).

4.4 MULTIPLATFORM ANALYSIS OF TEN CANCER TYPES

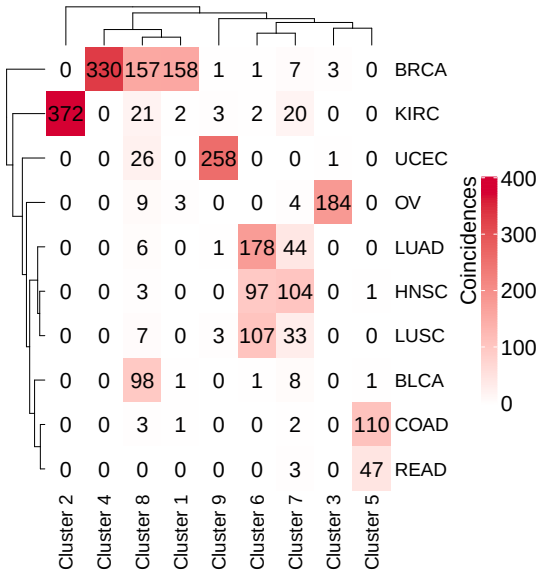
We apply the methodology developed in this chapter to the pan-cancer data presented in Chapter 3, combining the data layers both in the unsupervised and outcome-guided frameworks. We make use of the C implementation of MDI (Mason et al., 2016) to produce PSMs for each data layer separately. In order to be able to do so, we only include in our analysis the tumour samples that have no missing values; this reduces the sample size to 2,421 and the number of tumour types available for the analysis to ten. A mixture of Gaussians is used for the continuous layers (DNA copy number, miRNA, and protein expression), while the multinomial model is used for the methylation data, which are categorical. Due to the high number of features, it is not possible to produce a PSM for the full mRNA dataset, so we exclude it from the analysis presented here. In Appendix C, however, we show how the variable selection method developed in Chapter 2 can be employed in this case to reduce the size of each data layer and integrate all five data types.

Unsupervised integration

We combine the PSMs of the four data layers via multiple kernel k-means with number of clusters going from 2 to 50. We choose the number of clusters that maximises the silhouette, which is 9 (Appendix C). The resulting clusters are shown in Figure 4.9. Six out of the nine clusters contain almost exclusively samples from one tissue: most samples of renal cell carcinoma are in cluster 2, almost all statistical units in clusters 1 and 4 are breast cancer samples, most serous ovarian carcinoma samples are in cluster 3, bladder urothelial adenocarcinoma samples in cluster 8, and endometrial cancer samples in cluster 9. Cluster 5, instead, is formed by the colon and rectal adenocarcinoma samples together, and corresponds exactly to cluster 7 of Hoadley *et al.* Moreover, lung squamous cell carcinoma, lung adenocarcinoma, and head and neck squamous cell carcinoma are divided into two clusters. Cluster 8 contains the remaining samples. In Figure 4.10 are reported the average values of the silhouette when the number of clusters goes from 2 to 50. (the maximum is attained at $K = 15$) and the weights assigned to each PSM by the multiple kernel k-means algorithm. The average weights assigned to each data layer are: 6.9% to the copy number data, 7.5% to the methylation data, 7% to the miRNA expression data, and 78.7% to the protein expression data.

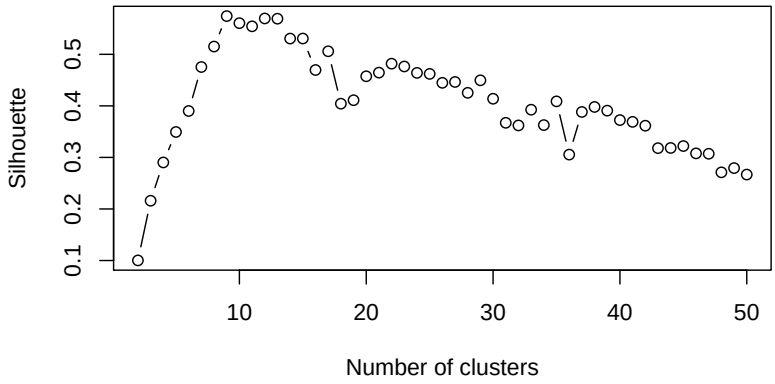


(A) Clusters and weighted kernel.

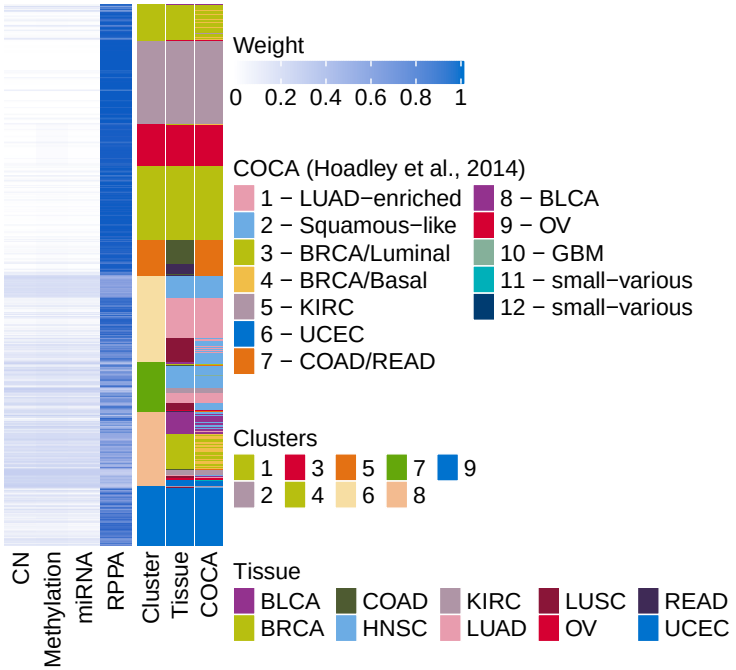


(B) Coincidence matrix.

FIGURE 4.9: Unsupervised multiplatform analysis of ten cancer types. (A) Left: weighted kernel. The rows and columns correspond to cancer samples. Higher values of similarity between samples are indicated in blue. Right: final clusters, tissues of origin, and COCA clusters. (B) Coincidence matrix comparing the tissue of origin of the tumour samples (rows) with the clusters (columns).



(A) Average silhouette.



(B) Weights.

FIGURE 4.10: Unsupervised multiplatform analysis of ten cancer types. (A) Average silhouette for number of clusters going from 2 to 50. (B) Weights assigned by the multiple kernel k-means algorithm to each observations in each layer, where "CN" stands for copy number and "RPPA" for reverse phase protein array.

Outcome-guided integration

We obtain the weights for the outcome-guided integration via the *SimpleMKL* algorithm, which are as follows: DNA copy number 35.9%, methylation 13.5%, miRNA expression 33.8%, and protein expression 16.8%. We then cluster the data using kernel k-means with number of clusters going from 2 to 50. The silhouette is maximised at $K = 27$ (Appendix C). The clusters obtained in this way are shown in Figure 4.12. It is interesting to note that, in this case, each cluster contains almost exclusively tumour samples from the same tissue. The only exceptions are clusters 4 and 22, which contain both lung and head/neck squamous cell carcinoma samples, and clusters 14 and 25 in which colon and rectal adenocarcinomas are clustered together, like in the unsupervised case. Each tumour type, except for ovarian and bladder cancers, is divided into multiple subclusters. Further analysis would be required to assess whether these clusters are clinically relevant. Interestingly, we observe a distinction between luminal (i.e. estrogen receptor-positive and HER2-positive) and basal breast cancer samples (the former are in clusters 8, 9, 17, 19, 23, 24, 27, while the latter are in cluster 13). This was also observed by Hoadley *et al*.

In Figure 4.11 are reported the average values of the silhouette when the number of clusters goes from 2 to 50. The maximum is at $K = 27$. Note that the clusterings and corresponding values of the silhouette may vary depending on the initialisation of kernel k-means. Due to this, slightly different clustering solutions may be found in different runs of the algorithm. However, we find that the results are quite consistent in this case.

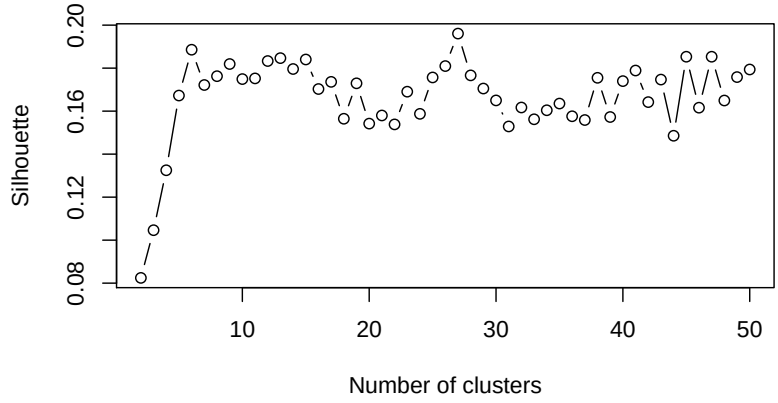
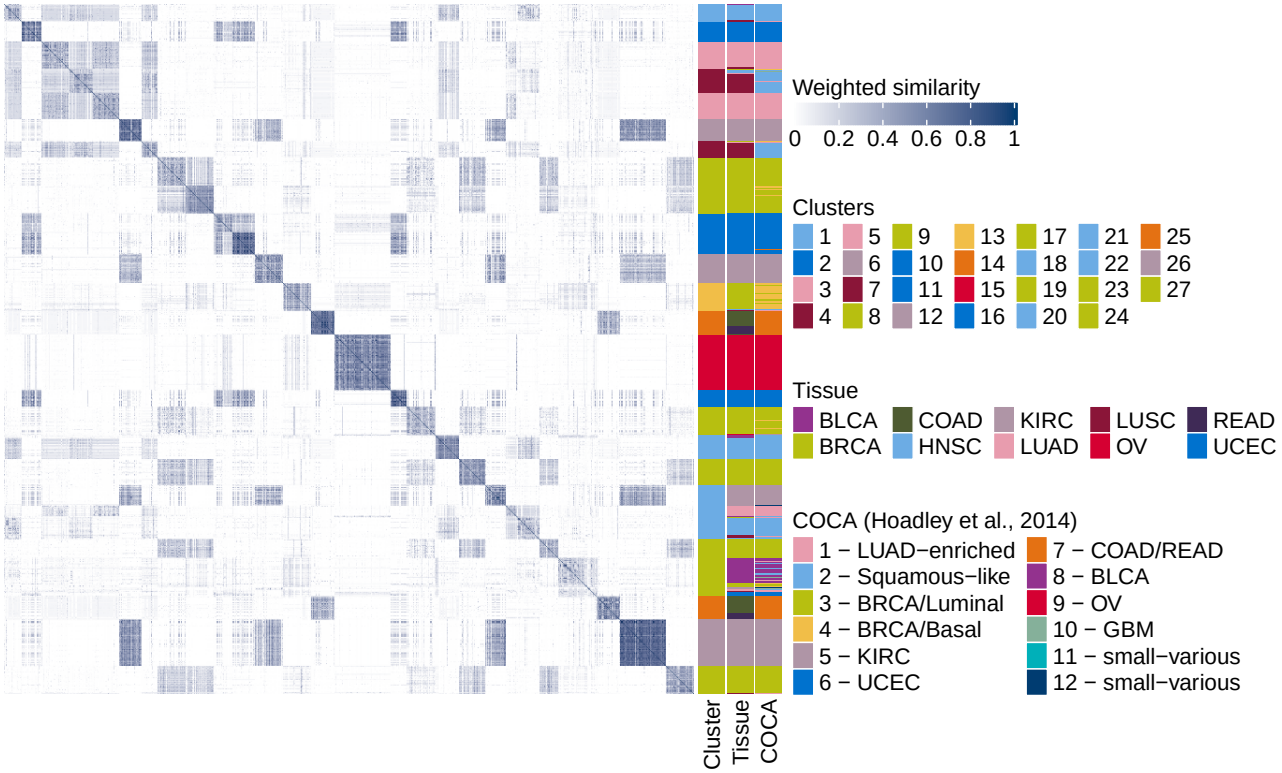
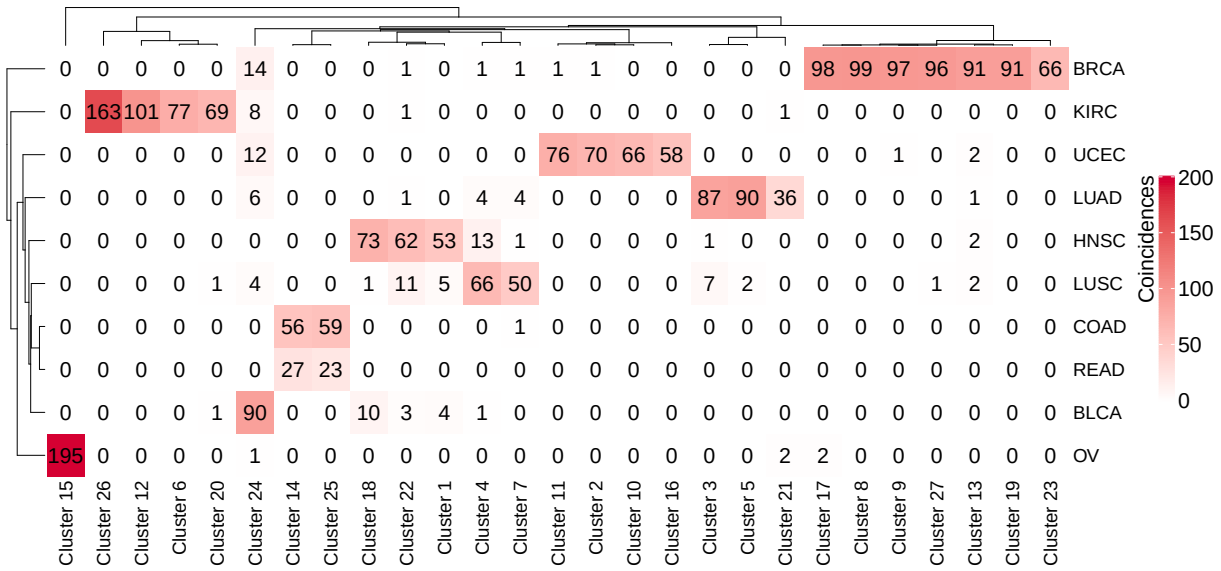


FIGURE 4.11: Outcome-guided multiplatform analysis of ten cancer types. Average silhouette for number of clusters going from 2 to 50.



(A) Clusters and weighted kernel.



(B) Coincidence matrix.

FIGURE 4.12: Outcome-guided multiplatform analysis of ten cancer types. (A) Left: weighted kernel. The rows and columns correspond to cancer samples. Higher values of similarity between samples are indicated in blue. Right: final clusters, tissues of origin, and COCA clusters. (B) Coincidence matrix comparing the tissue of origin of the tumour samples (rows) with the clusters (columns).

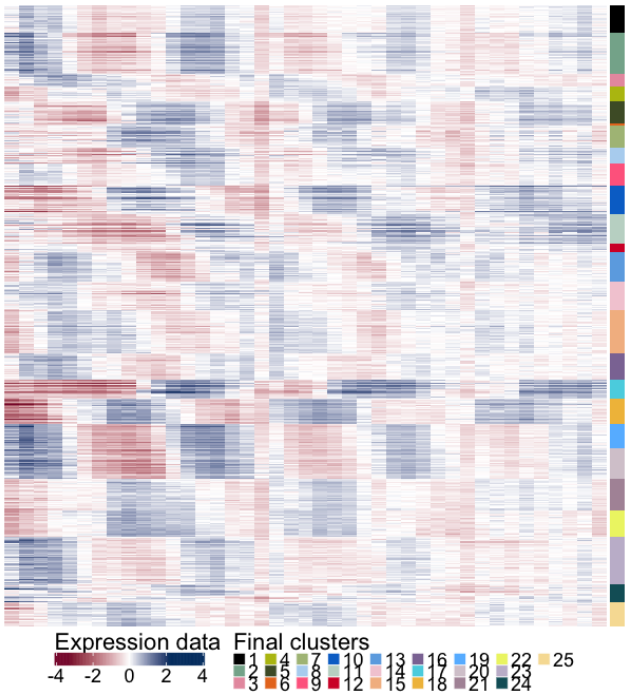
4.5 TRANSCRIPTIONAL MODULE DISCOVERY

We now revisit the transcriptional module discovery example of Chapter 3. To produce the PSMs for the two datasets, we use the `DPMSysBio` Matlab package of Žurauskienė, Kirk, and Stumpf (2016). For each dataset, we run 10,000 iterations of the MCMC algorithm and summarise the output into a PSM. The PSMs obtained in this way are reported in Appendix C.

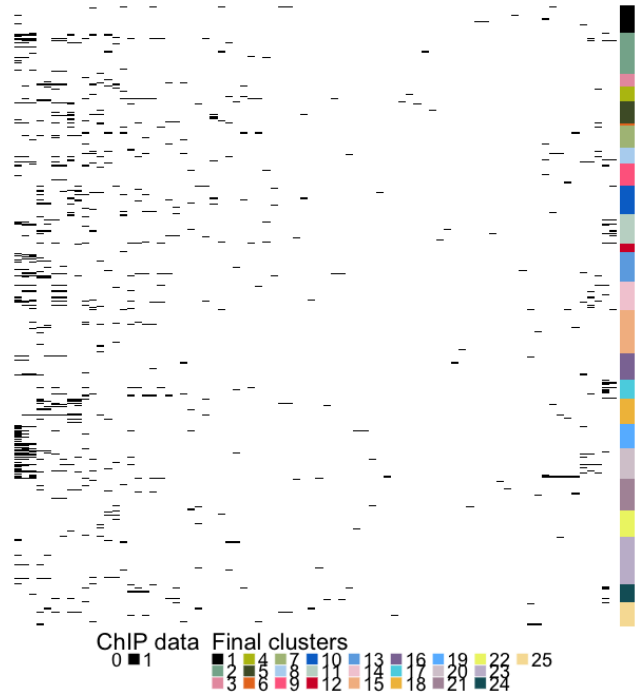
We combine the PSMs using KLIC. The average weights assigned by KLIC to each PSM and the values of the average silhouette for different numbers of clusters are reported in Appendix C. We set the number of clusters to 25, which is the value that maximises the silhouette. The final clusters are shown in Figure 4.13 next to the two datasets and the combined PSM. In order to determine whether our clustering is biologically meaningful, we use the GOTO scores defined in Chapter 3, which are reported in Table 4.1. The two datasets combined achieve higher GOTO scores than those of the clusters obtained using each dataset separately. We also compare these GOTO scores to those obtained with the two methods used in Chapter 3: COCA and KLIC used to combine kernels derived from CC. Both have lower GOTO scores than the novel approach presented in this chapter. For COCA, this is result not unexpected, since the method is unweighted and has previously been shown to perform less well than KLIC. The difference between the two variants of KLIC only lies in how the kernels are constructed. These scores therefore suggest that kernels generated from probabilistic models can lead to more accurate results than those built using consensus clustering.

Dataset(s)	GOTO BP	GOTO MF	GOTO CC
ChIP data	6.18	0.97	8.54
Expression data	7.07	1.04	8.90
ChIP+Expression data: COCA	5.74	0.90	8.19
ChIP+Expression data: CC + KLIC	6.60	0.96	8.66
ChIP+Expression data: PSM + KLIC	**7.15**	**1.05**	**8.93**

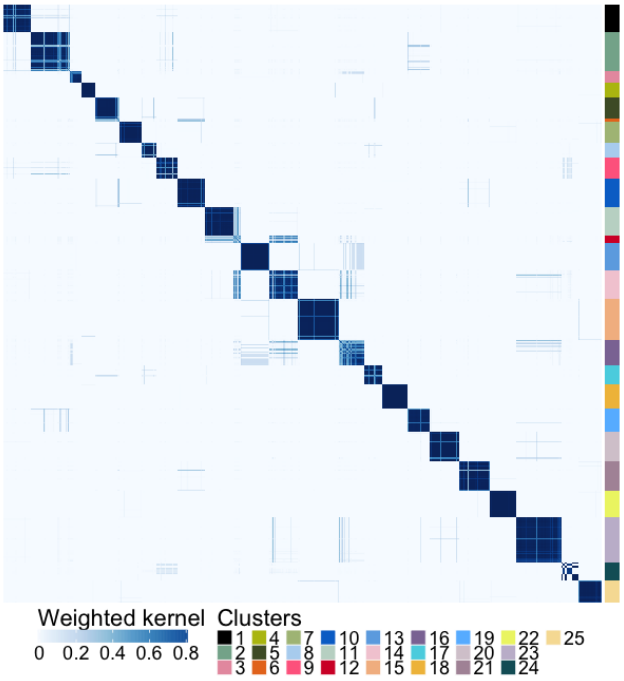
TABLE 4.1: Gene ontology term overlap scores. BP stands for Biological Process ontology, MF for Molecular Function, and CC for Cellular Component. The number of clusters used for every method is 25.



(A) Expression data.



(B) ChIP-chip data.



(C) Weighted kernel.

FIGURE 4.13: Transcriptional module discovery, integration of the Harbison et al. (2004) and Granovskaia et al. (2010) datasets. (A) Expression data. Each row corresponds to a gene and each column to a different time point. (B) ChIP-chip data. Each row corresponds to a gene and each column to a transcriptional regulator. (C) Weighted kernel. The rows and columns correspond to the genes. Higher values of similarity between genes are indicated in blue. To the left of each plot is shown the final clustering, obtained by integrating the PSMs of the expression and ChIP-chip data via multiple kernel k-means.

4.6 DISCUSSION

The main findings of this chapter are summarised here. We also describe some challenges encountered, which open new research questions to be explored in the future.

4.6.1 *Main findings*

We have presented a novel method for summarising a sample of clusterings from the posterior distribution of an MCMC algorithm for Bayesian clustering, based on kernel methods. We have also extended this method to allow us to integrate multiple PSMs. This can be done either in an unsupervised or in an outcome-guided way. The former weights each PSM according to how well defined is the clustering structure that it describes. The latter gives more importance to the PSMs that assign higher similarities to the groups of observations that belong to the same class, according to the response variable of choice; this allows to uncover more meaningful partitions of the data, and simultaneously find out which PSMs (or kernels, more in general) are useful for the application at hand.

We have used simulation examples to show that our method gives comparable performances in terms of proportion of correct co-clustering as existing techniques when just a single dataset is being clustered, in the unsupervised setting. We have also demonstrated that the integration of multiple datasets gives better results than using one dataset at a time, if they all describe the same clustering structure. In situations where the clustering structure is not the same in all datasets, the outcome-guided version of KLIC can be used, if a (categorical) response variable related to the output of interest is available, to assign higher weights to the PSMs defining partitions of the data similar to the one described by the response. The simulation examples demonstrate that this feature can be extremely useful when not all the PSMs have the same clustering structure.

Finally, we have applied the novel methods to two real data applications. The pan-cancer data analysis shows that the outcome-guided integration of multiple PSMs can potentially be used in the context of tumour subtype discovery. The yeast example demonstrates that the proposed method is able to identify groups of genes that are co-expressed and co-regulated that are more biologically meaningful than those determined via state-of-the-art integrative algorithms.

4.6.2 *Challenges*

In this chapter we have overcome the issue of choosing the number of clusters encountered in Chapter 3 making use of DPMMs, that do not require knowing the value of K. However, evaluating clustering results remains a complex task,

and two new questions arise as a result of the analyses presented in this chapter, which we now discuss.

Mixing of the MCMC chains

Implementing MCMC schemes for DPMMs that work on large datasets is not a trivial task. The most evident problem is that MCMC is computationally demanding. For the mRNA dataset available for the pan-cancer analysis, for instance, running the C implementation of MDI on Cambridge University's high-performance computing cluster would have taken a prohibitively long time. Another, less apparent issue, is the mixing of the MCMC chains. Again, this is easily observable in the cancer subtyping applications. The figures reported in Appendix C used to assess MCMC convergence show very poor mixing: in most cases, even if the cluster allocations differ slightly at each iteration, the number of clusters remains constant. This means that the parameter space has not been explored correctly, and successive samples are not independent. Although not the focus of this work, this motivates the use of more scalable sampling schemes for problems of this type. This is an area of active research; some recent publications on scalable inference for DPMMs via MCMC include those of Chang and Fisher III (2014), Ge et al. (2015), and Ni et al. (2020).

Comparing partitions

We have seen in the simulation studies reported in Section 4.3 and in the analysis of the pan-cancer data in Section 4.4 that the outcome-guided version of KLIC can identify subclusters in the data. In multi-omic applications the standard way of comparing partitions of the data is to use the ARI (see e.g. Kirk et al., 2012; Lock and Dunson, 2013; Gabašová, Reid, and Wernisch, 2017). This index only indicates whether two partitions are similar overall or not, but is not suitable to compare two partitions of the data in situations where one of the two subdivides one or more classes into smaller subsets. A different way of assessing the similarity of two clusterings needs to be developed for this.

5

INTEGRATIVE CLUSTERING OF MULTI-OMIC CARDIOMETABOLIC SYNDROME DATA

In this chapter, we continue the CMS data analysis of Chapter 2, applying the unsupervised and outcome-guided integration algorithms introduced in Chapters 3 and 4 respectively to it. The goal here is to find clusters of individuals who have similar features and check whether any blood donors belong to the same clusters as some of the CMS patients. This may help identify seemingly healthy individuals who may be at risk of developing CMS.

As we have seen, the CMS dataset has a large number of missing values. Therefore, we first need to explain how KLIC can cope with incomplete datasets.

The data analysis presented here provides an example of why variable selection is of fundamental importance in cluster analysis (Fop and Murphy, 2018); especially for high-dimensional data. In particular, we show that if DPMMs are fitted on the full CMS 'omic layers, it becomes impossible to pick out the clustering structure in each layer that separates people affected by CMS from the others. This is because the large number of irrelevant variables in each layer, which have either different or no clustering structure, "hide" the signal that we are interested in. Additionally, we show that current MCMC algorithms are prohibitively slow and get stuck in local modes, when dealing with such high-dimensional datasets. However, using only the features selected via penalised logistic regression in Chapter 2, we are able to fit DPMMs on each layer and integrate them using our MKL approach to find meaningful clusters. This shows that variable selection improves not only the interpretability of the model, but also the quality of the final clusters.

Chapter outline

First, the unsupervised and outcome-guided KLIC algorithms are extended in order to analyse incomplete multi-omic datasets in Section 5.1. Then, both methods are applied to the CMS data in Section 5.2. The main findings and challenges of this chapter are summarised in Section 5.3.

5.1 HANDLING MISSING DATA

We have seen in Chapter 2 that multi-omic datasets often have missing observations. This section is dedicated to giving further details about how missing data

can be handled by using unsupervised and outcome-guided KLIC. For simplicity, throughout this chapter we consider the data of layer m to be missing for individual n even if only one of the features is missing. In this way, we can easily extend the algorithms of unsupervised and outcome-guided KLIC to the case of incomplete data. More sophisticated techniques would have to be developed in order to define similarities between incomplete 'omic measurements that utilise all available data, however this falls outside of the scope of this chapter.

5.1.1 *Unsupervised KLIC*

The optimisation problem that is solved to find the optimal clustering and weights in localised multiple kernel k-means is:

$$\begin{aligned} \underset{H,\Theta}{\text{maximise}} \quad & \operatorname{tr}(H'\Delta_\Theta H) - \operatorname{tr}(\Delta_\Theta) \\ \text{subject to} \quad & H'H = \mathbb{I}_K, \\ & \Theta' 1_M = 1, \\ & \Delta_\Theta = \sum_m (\boldsymbol{\theta}_m \boldsymbol{\theta}_m') \circ \Delta_m, \end{aligned} \tag{5.1a}$$

where $\circ$ is the Hadamard product. As stated in Chapter 3, one can optimise the objective function of Equation (5.1a) with a two-step procedure, that iteratively (i) solves a standard kernel k-means problem with kernel δ_Θ, keeping the weight matrix Θ fixed and then (ii) optimises the objective function with respect to Θ. Again, the first step reduces to solving one optimisation problem with a single kernel and in the second step one just needs to solve a QP problem (Gönen and Margolin, 2014). In particular, the QP problem in step (ii) is:

$$\begin{aligned} \underset{\Theta}{\text{minimise}} \quad & \sum_{m=1}^{M} \boldsymbol{\theta}_m^T \left[(\mathbb{I}_N - HH^T) \circ \Delta_m \right] \boldsymbol{\theta}_m \\ \text{subject to} \quad & \Theta \in \mathbb{R}_+^{N \times M}, \\ & \Theta' 1_M = 1_N. \end{aligned}$$

We now extend the formulation of Gönen and Margolin (2014) so that clustering can be performed even if not all kernels include information for all pairs of observations. We do so by making sure that each kernel matrix is of size $N \times N$, where the rows and columns corresponding to the observations that do not have complete information available in the corresponding data layer are filled with zeros. More precisely, we define by $I_m \subset \{1, \ldots, N\}$ the set of the missing values in each dataset $m = 1, \ldots, M$ and make sure that the corresponding kernel Δ_m is

such that

$$\Delta_{ij}^m = 0 \quad \forall i \in I_m, j \neq i,$$
$$\Delta_{ii}^m = 1 \quad \forall i \in I_m.$$

The resulting matrix Δ_m is a weighted sum of co-clustering matrices with structure

$$\Delta_m = \begin{bmatrix} \Delta'_m & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where Δ'_m is the mth kernel matrix for the available data and the observations are ordered such that the missing ones are at the bottom of the matrix for presentational purposes. Therefore, it is a valid kernel matrix.

Moreover, it is possible to cancel the influence of the missing observations on the final solutions by setting their weight to zero in optimisation problem (5.2):

$$\begin{aligned} \underset{\Theta}{\text{minimise}} \quad & \sum_{m=1}^{M} \boldsymbol{\theta}_m^T \left[(I_n - HH^T) \circ \Delta_m \right] \boldsymbol{\theta}_m \\ \text{subject to} \quad & \Theta \in \mathbb{R}_+^{N \times M}, \\ & \Theta' 1_M = 1_N, \\ & \theta_{mi} = 0 \qquad \forall i \in I_m,\ m = 1, \ldots, M. \end{aligned} \tag{5.3a}$$

This corresponds to adding $|I_1| + \cdots + |I_M|$ equality constraints, each one on a different variable, or, equivalently, to removing a number $|I_1| + \cdots + |I_M|$ of variables from the optimisation problem. Therefore, (5.3) is a QP problem.

The objective function (3.8) can then be minimised by iterating between steps (i) and (ii) as in the previous case, with the additional constraints (5.3a) in step (ii).

5.1.2 *Outcome-guided KLIC*

The outcome-guided KLIC algorithm only outputs one overall weight for each data layer. Therefore, the strategy used for unsupervised KLIC cannot be applied to this case. Instead, we infer the kernel weights using only the statistical units that have no missing observations. Then, we define a weight matrix $\Theta \in \mathbb{R}^{N \times M}$ containing sample-specific weights, as follows:

- for the observations that have no missing values, the weight of each layer corresponds to the weight assigned by the SVM to that layer;
- for the remaining statistical units, the weights of the incomplete or missing

layers are assigned zero weight, whereas the others have the SVM weights, normalised in order to sum to one.

The weighted kernel is then obtained in the same way as in unsupervised KLIC, i.e.

$$\Delta_\Theta = \sum_m (\boldsymbol{\theta}_m \boldsymbol{\theta}'_m) \circ \Delta_m,$$

where θ_m, $m = 1, \ldots, M$, are the columns of Θ.

5.2 APPLYING KLIC TO THE CARDIOMETABOLIC SYNDROME DATA

We now apply unsupervised and outcome-guided KLIC to the CMS dataset presented in Chapter 2.

In order to apply KLIC to the CMS data, we want to use the MDI implementation in the C programming language of DPMMs, introduced in Chapter 4, to build a PSM for each data layer. However, we found that the MCMC chains do not reach convergence on the full datasets (for further details see Appendix D). For this reason, we only include in the models the variables selected via separate EN in Chapter 2, which constitute the molecular signature of CMS. The convergence assessment for the MCMC chains and the resulting PSMs are reported in Appendix D.

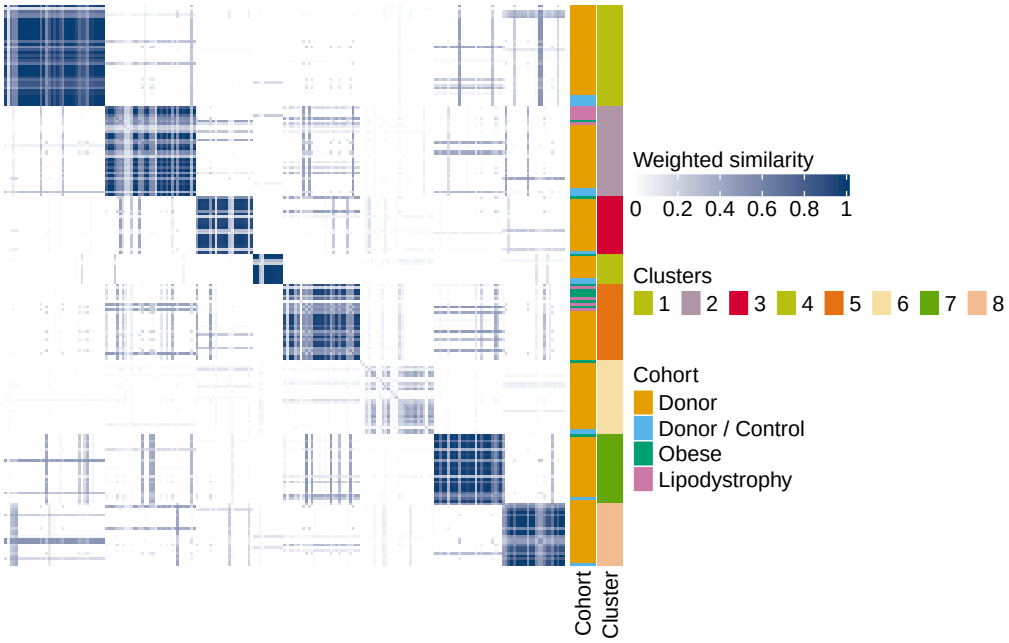
The importance of variable selection in model-based clustering has been highlighted by numerous publications; for a review see Fop and Murphy (2018). In the context of integrative 'omics, in particular, we mentioned in Chapter 3 that the iCluster integrative method (Shen, Olshen, and Ladanyi, 2009) has been extended to perform variable selection (Kim et al., 2017). Variable selection for DPMMs has also been the object of several publications (e.g. Tadesse, Sha, and Vannucci, 2005; Kim, Tadesse, and Vannucci, 2006).

Fop and Murphy (2018) divide variable selection methods for clustering algorithms into two categories: *filter methods* carry out the variable selection step separately from the clustering, while in *wrapper methods* clustering and variable selection are performed jointly. The former are easier to implement and less computationally intensive, the latter have been shown to give better results. The approach followed here falls into the first category of methods. Given the high dimensionality of the data, performing variable selection via penalised logistic regression as a pre-filtering step is extremely advantageous: it improves both the quality of the inference and lowers its computational cost.

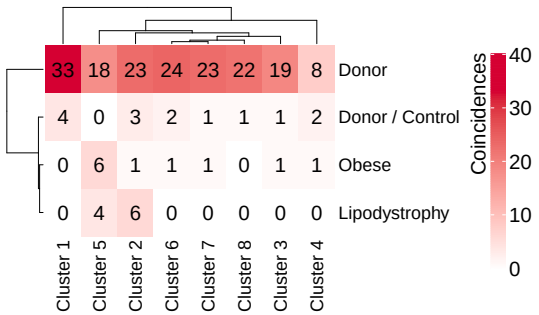
5.2.1 *Unsupervised integration*

First, we perform unsupervised integration on the eight PSMs obtained for the CMS data. Figure 5.1 shows the weighted kernel and final clusters, as well as a comparison of the clusters to our classification of the individuals in the study into donors, controls, lipodystrophy and obese individuals. In Figure 5.2 are reported the average values of the silhouette when the number of clusters goes from 2 to 45 (the maximum is attained for $K = 8$) and the weights assigned to each PSM by the multiple kernel k-means algorithm. The average weights assigned to each layer are as follows: ChIP-seq monocytes 15.31%, ChIP-seq neutrophils 1.45%, RNA-seq monocytes 2.33%, RNA-seq neutrophils 12.28%, methylation monocytes 31.06%, methylation neutrophils 1.98%, metabolites 2.19%, lipids 33.40%.

Interestingly, even though we built the PSMs using the variables that discriminate the obese individuals from the control donors, the lipodystrophy patients occupy only two clusters, one of them containing more than half of the obese individuals. This seems to confirm once again that the two extreme phenotypes considered in this study may have some commonalities on the molecular level.

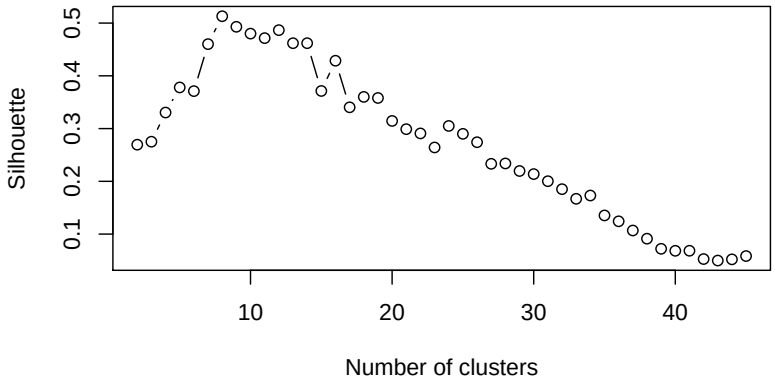


(A) Clusters and weighted kernel.

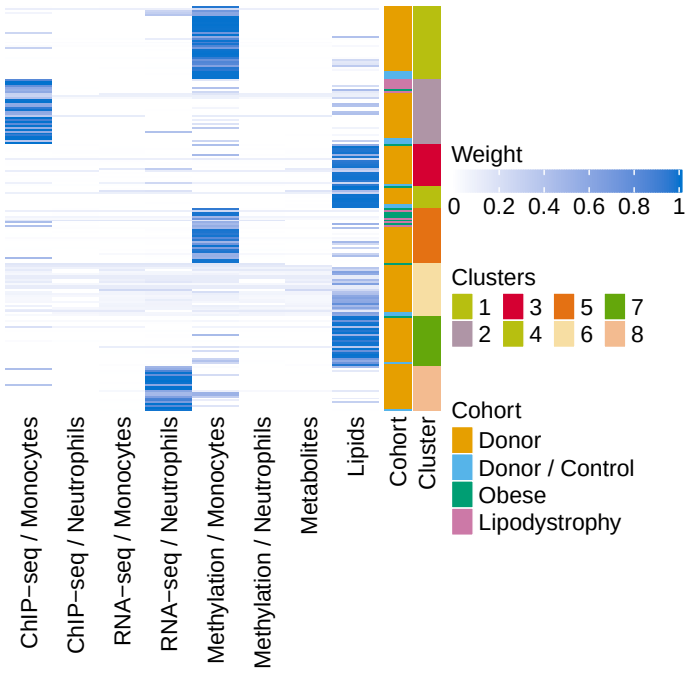


(B) Coincidence matrix.

FIGURE 5.1: Unsupervised integration of the CMS data. (A) Left: weighted kernel. The rows and columns correspond to individuals in the study. Higher values of similarity between samples are indicated in blue. Right: cohorts and final clusters. (B) Coincidence matrix comparing the cohort of each individual (rows) with the clusters (columns).



(A) Average silhouette.



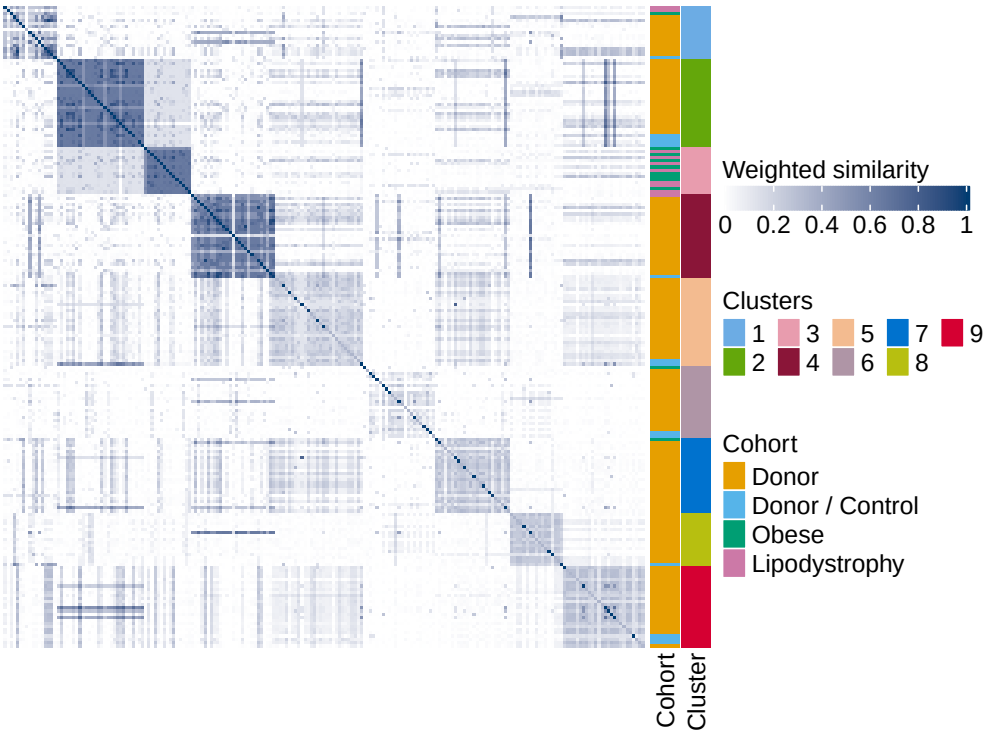
(B) Weights.

FIGURE 5.2: Unsupervised integration of the CMS data. (A) Average silhouette for number of clusters going from 2 to 45. (B) Weights assigned by the multiple kernel k-means algorithm to each observation in each layer.

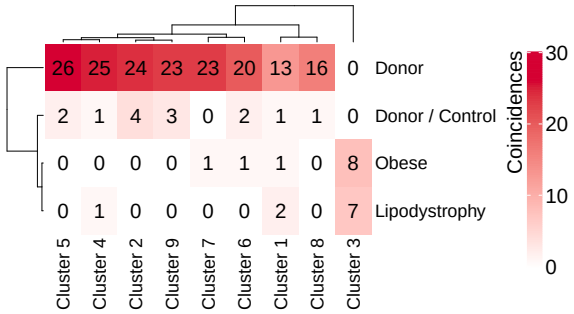
5.2.2 *Outcome-guided integration*

We now use the same PSMs as in the previous section as input to outcome-guided KLIC. As a response, we use a binary variable that is equal to one for individuals who belong to one of the two extreme phenotype groups, and is equal to minus one for the blood donors. Note that we have already used the cohort information in the variable selection step, and therefore we cannot exploit this example to evaluate the quality of the inference obtained via outcome-guided KLIC. Figure 5.3 shows the weighted kernel and final clusters, and a comparison between the clusters and cohorts. In Figure 5.4 are reported the average values of the silhouette when the number of clusters goes from 2 to 45 (the maximum is attained for $K = 9$) and the matrix of weights. The weights assigned by the simpleMKL to each data layer are as follows: ChIP-seq monocytes 6.17%, ChIP-seq neutrophils 43.94%, RNA-seq monocytes 29.95%, RNA-seq neutrophils 8.13%, methylation monocytes 5.50%, methylation neutrophils 0.35%, metabolites 4.04, lipids 1.92%.

As expected, most of the extreme phenotype individuals belong to the same cluster. It is also interesting to note that the individuals in cluster 2 have consistently higher similarities with the extreme phenotype cluster (number 3) than the remaining ones. This might suggest that the donors in cluster 2 should be analysed in greater detail to ascertain if they have CMS or are at greater risk of developing it.

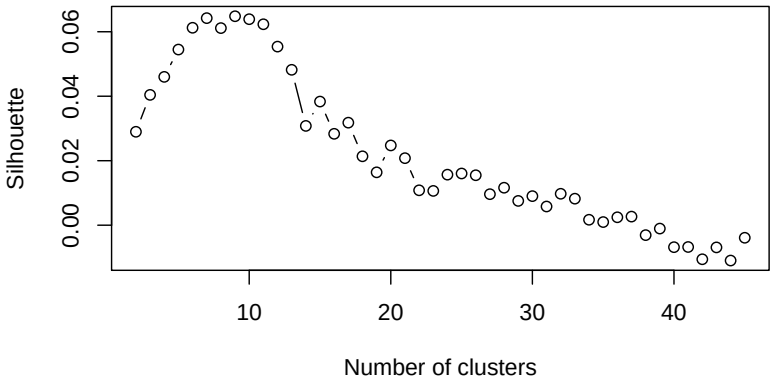


(A) Clusters and weighted kernel.

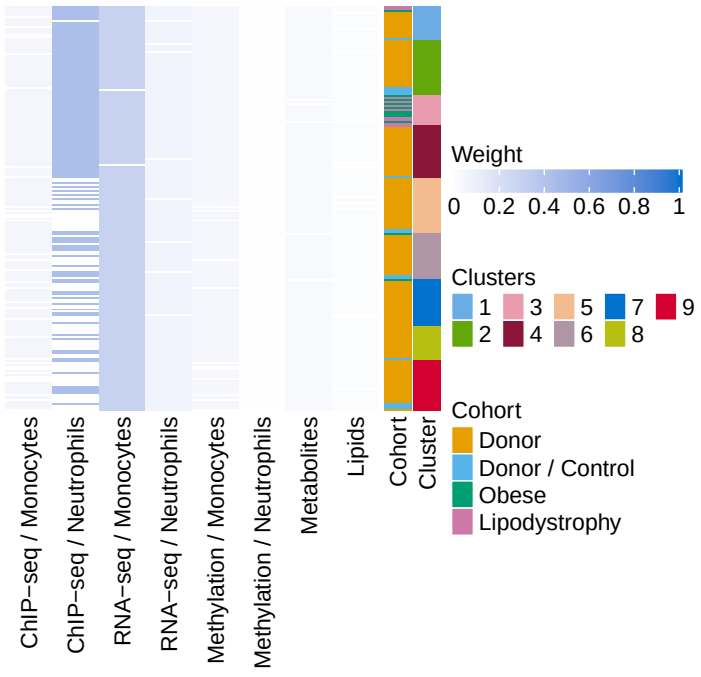


(B) Coincidence matrix.

FIGURE 5.3: Outcome-guided integration of the CMS data. (A) Left: weighted kernel. The rows and columns correspond to individuals in the study. Higher values of similarity between samples are indicated in blue. Right: cohorts and final clusters. (B) Coincidence matrix comparing the cohort of each individual (rows) with the clusters (columns).



(A) Average silhouette.



(B) Weights.

FIGURE 5.4: Outcome-guided integration of the CMS data. (A) Average silhouette for number of clusters going from 2 to 45. (B) Weights assigned by the algorithm to each observation in each layer.

5.3 DISCUSSION

In this chapter, we have seen how the statistical methods proposed in this thesis can handle missing data and be applied to a large multi-omic dataset such as the novel CMS dataset presented in Chapter 2. We summarise here the main findings and challenges of this chapter.

5.3.1 *Main findings*

After using penalised logistic regression to identify a molecular signature of CMS in each 'omic layer, we were able to build a PSM for each layer using MCMC schemes for DPMMs. This allowed us to apply unsupervised and outcome-guided KLIC to this dataset, after showing that both algorithms can easily incorporate information from statistical units that have incomplete data.

The results obtained in this chapter illustrate the difference between unsupervised and outcome-guided approaches. As we have seen in Chapters 3 and 4, most integrative algorithms for the stratification of multi-omic data are unsupervised. Only a few methods have been recently developed which exploit a response variable for clustering purposes (see Arora et al., 2020). However, the work presented in this chapter shows how very different clusterings can be obtained using the same input data (in this case, the same PSMs), depending on whether the integration is unsupervised or outcome-guided. It is important to note, moreover, that the DPMMs were fitted using the variables representing the molecular signature of CMS: this was not sufficient to make the unsupervised integration find a partition that differentiates between extreme phenotype individuals and the others. In general, different clustering structures can be found in rich datasets like the one used here and, even if unsupervised KLIC has a meaningful way of assigning weights to each PSM, doing so in an unsupervised way does not guarantee to find the most relevant clustering structure for the problem at hand. Conversely, making use of a response variable that is associated to the population structure that is to be uncovered can dramatically improve the usefulness of the data partition found. In situations where response variable is not available, it is still crucial to be aware that different weightings of the 'omic layers can be considered, each yielding a different output, and to be able to monitor the impact of each data layer on the final clustering. This suggests that the results obtained using clustering methods that do not explicitly determine optimal weights for each data layer (e.g. COCA; The Cancer Genome Atlas Research Network, 2012) or do not take into consideration a response variable (e.g. iCluster, Clusternomics, and even unsupervised KLIC; Shen, 2012; Gabašová, Reid, and Wernisch, 2017; Cabassi and Kirk, 2020b) might be limited.

5.3.2 *Challenges*

The methods and results presented in this chapter could be improved and extended in two main directions, which we explain here.

Loss of information in outcome-guided KLIC

In the first step of outcome-guided KLIC, where the kernel weights are defined, only the statistical units that have complete measurements on each data layer are used. Therefore, information from incomplete observations is discarded at this step. It would be useful to develop a way to include information relative to all the available data at this stage of the analysis.

Interpretation of the results

Further analysis and expert knowledge would be required to gain a deeper understanding of the two partitions of the data that we have identified here. In particular, it would be interesting to determine whether the individuals belonging to cluster 2 in the outcome-guided analysis are affected, at some level, by CMS.

6

DISCUSSION

We present here the main findings of this thesis (Section 6.1), some possible future research areas (Section 6.2) and concluding remarks (Section 6.3).

6.1 MAIN FINDINGS

The flow of genetic information within a cell and its influence on phenotype is complex, and yet to be fully understood. As opposed to more traditional approaches such as, for example, GWAS, multi-omic studies take an holistic approach and aim at disentangling the interactions between multiple types of 'omics and understanding their effect on many aspects of an organism's life. The importance of 'omic studies combining different layers of information has been highlighted by several high-profile publications (Hasin, Seldin, and Lusis, 2017; Karczewski and Snyder, 2018). In this thesis, we have developed novel statistical methods that can be used to combine and extract information from multi-omic datasets, and have used them to draw meaningful biological and clinical information from three real-world datasets. Compared to existing approaches, the methods developed here put more emphasis on monitoring how much impact each data layer has on the final output, while ensuring that the corresponding algorithms are not computationally prohibitive for very large datasets, such as those that are often encountered in practice.

6.1.1 *Supervised integration*

The supervised integration of multi-omic data can be used to learn predictive and classification models in a wide range of applications. In this thesis, we have focused on binary classification, taking as an example the classification of patients according to their disease status. 'Omic datasets are usually high-dimensional, with more covariates than data points. For this reason, penalised likelihood approaches are often used to build predictive models for this type of data. The challenge is then how to apply these methods to multi-omic datasets and/or extend them methods while taking into account the fact that different 'omic layers have different characteristics (e.g. level of noise, number of covariates, percentage of relevant covariates). Throughout this thesis, we have also put emphasis on the importance of variable selection in the supervised integration of multi-omic data.

Identifying a subset of relevant features for the problem at hand not only makes predictive models more interpretable, but is also useful for other downstream analyses (e.g. clustering).

In the first part of Chapter 2, we introduced two ways of building binary classification models for large multi-omic datasets, while selecting a set of covariates that are jointly predictive of the outcome of interest. Both approaches are composed of two steps: first, variables are selected in each 'omic layer, then all the selected variables are combined into a joint model. The difference between the two is that, in one case, the user can manually pick the penalty parameters, depending on how many variables they want to select. In the other one, the penalty parameters are selected automatically in order to minimise the CV error. In both cases, the MR rates observed in the simulation studies are comparable to those of state-of-the-art competitor methods. At the same time, the number of relevant variables identified by the two-step approaches is higher than for the other methods, indicating that the former are more suitable for multi-omic signature identification.

The supervised approaches to multi-omic integration developed in this thesis will enable other researchers to build predictive models for large multi-omic datasets. This could help build binary classification models and identify sets of relevant 'omic features in a variety of multi-omic studies. To make this easier, we have made our code for our novel approaches publicly available, and have extended the functionalities of the main competitor method to the logistic regression framework.

6.1.2 *Unsupervised integration*

In applications where the goal is to find clusters of individuals or genes that share similar characteristics across multiple 'omics, unsupervised integration is often used. Statistical methodologies that combine multiple 'omic layers in an unsupervised fashion are currently being produced at a fast pace (Rappoport and Shamir, 2018). However, the objective function that is optimised by these methods is not always clear. Another challenge of unsupervised clustering is how to determine to what extent each data type should impact the final clustering, given that the level of noise and cluster separability can greatly vary across layers and that different layers may contain contrasting information. For most unsupervised integrative algorithms for multi-omic data, even determining a posteriori which data layer influenced the final output the most is often an arduous task. A notable example of unsupervised integrative clustering algorithm suffering from these problems is COCA, a widely-used tool in cancer applications (The Cancer Genome Atlas Research Network, 2012).

In Chapter 3, we laid out for the first time the algorithm behind COCA and systematically explored its properties. We then proceeded to suggest a novel method for the unsupervised integration of multiple 'omics based on multiple kernel learning, named KLIC. Contrary to existing approaches, the KLIC algorithm explicitly assigns a weight to each omic layer, up-weighting the most informative ones. We showed that KLIC can be run on large datasets and that the clusters found by KLIC more accurately reflect the true clustering structure than those of the main competitor methods, across a range of simulation settings. This happens for two main reasons: firstly, the main competitor methods assign too much weight to noisy datasets; secondly, the way in which kernels are built in KLIC makes it more robust than the other methods to the inclusion of noisy variables.

The version of KLIC used in Chapter 3 employs consensus matrices as kernels. However, we also pointed out that any other PSD matrix whose entries represent similarities between the statistical units can be used. This turned out useful in Chapter 4, where clever ways to summarise and combine PSMs were sought. Indeed, after proving that PSMs are PSD matrices, we were able to use the KLIC algorithm to summarise the output of MCMC schemes run on each data layer independently. Thanks to this, we also managed to overcome the problem of choosing the number of clusters when building the kernel matrices. In fact, we explained how to build PSMs (and therefore kernel matrices) from DPMMs, which do not require fixing the value of K.

We also showed that, having proven that PSMs are valid kernel matrices, kernel methods can be applied to them to summarise the output of just one MCMC sample with the same accuracy as the other methods from the literature.

The unsupervised version of KLIC will allow researchers to perform unsupervised integrative clustering of multi-omic data in a principled way, defining similarities in between observations based on either heuristic clustering algorithms or model-based techniques, depending on the user's needs, and assigning a different weight to each data layer. It will also allow to include external information, as long as it can be put in the form of a PSD matrix. The R packages `coca` and `klic` have been designed to make this as simple as possible.

6.1.3 *Outcome-guided integration*

It has recently become evident that high-dimensional 'omic datasets can be used to define multiple partitions of the data. For this reason, it is essential to determine at the beginning of the analysis what is the purpose of clustering in the application at hand and what type of partition is sought. In order to pick out the most relevant population structure, a response variable that is known to be related to the outcome of interest can be used to guide the clustering. In the context of multi-omic integration, this has been done for example by Arora et al. (2020),

who developed an outcome-weighted strategy for cancer patients stratification, making use of survival information to guide the clustering.

In Chapter 4, KLIC is extended to take into account a categorical response variable. To this end, SVMs are used to define the kernel weighting, so that the kernels presenting a clustering structure similar to the partition induced by the response are up-weighted. We demonstrated through simulation studies that this approach is particularly useful in two situations. Firstly, if one of the layers has a clear clustering structure that is unrelated to the response variable, the corresponding kernel matrix is down-weighted. Secondly, this approach can help uncover a more refined partition of the data than the one reported by the response variable.

We have shown in Chapter 5 that, with the same input, the unsupervised and outcome-guided versions of KLIC can produce completely different outputs. This suggests that, whenever possible, a response variable should be used to partition multi-omic datasets. Therefore, outcome-guided KLIC could be used in the future to make sure that gene and patient stratification are performed in a meaningful way.

6.1.4 *Real data applications*

While the contributions of this thesis are mainly methodological, the three real data applications presented in this thesis also generated interesting insight.

Cardiometabolic syndrome data

The first real-world application presented in this thesis was aimed at finding a molecular characterisation of CMS, a set of metabolic dysfunctions that are known to be associated with higher risk of type 2 diabetes and CVD. For this, eight 'omic layers were available, for a set of 184 blood donors and 21 individuals affected by this syndrome.

The second part of Chapter 2 is dedicated to the supervised analysis of the CMS dataset, which is divided into three steps.

- *Multivariate signature identification.* The two-step logistic regression methods introduced in the first part of Chapter 2 are used to identify a molecular signature in each layer and to subsequently fit a ridge-penalised logistic regression model. The predicted probabilities of each individual to belong to the group of CMS patients showed that obese and lipodystrophy individuals have similar molecular profiles; this seems to validate the hypothesis that different types of CMS have common molecular pathways. Moreover, the predictive model could also be used in the future to identify blood donors

who have a high probability of having CMS for prevention and diagnostic purposes.

- *Univariate differential analysis.* We compared our molecular signatures to those obtained by a univariate approach often used by practitioners. As expected, only a few features were selected by the univariate approach, due to the fact that the multiple testing correction makes the selection criterion quite stringent, and that it is unable to identify synergies between different features that are together predictive of CMS.
- *Validation via external cohorts.* The analysis of the Fenland cohort validated our lipidomics signature of CMS. Additionally, the NASH cohort showed a similar pattern of associations between known CMS risk factors and the selected lipids to the one observed in our novel dataset.

Overall, our multi-omic analysis allowed us to identify a set of molecular features that can help discriminate between extreme phenotype and healthy individuals, with greater accuracy than standard univariate approaches. Further results reported in Appendix A validate our conclusions. Once again, we reiterate that these are only explorative analyses; a larger number of observations must be collected and analysed before using these results in the clinic.

In Chapter 5, unsupervised and outcome-guided KLIC are applied to the CMS dataset, to show the importance of variable selection in clustering, and to demonstrate how unsupervised and outcome-guided KLIC can handle missing data. The analysis of the CMS data presented in Chapter 5 also constitutes an example of why it is important to use a response variable to guide the clustering algorithm. Indeed, different clustering structures can be found in the CMS data, even after selecting the variables that are predictive of patient status via penalised logistic regression. Therefore, using a response to guide the kernel weighting allows us to find the most relevant one.

Multiplatform analysis of 12 cancer types

Multi-omic clustering methods have been widely applied to cancer studies, yielding many interesting results. Most notably, these have been used to identify new cancer subtypes, which can help develop more effective and less invasive treatment for each novel subtype.

Here, we focused in particular on reproducing the clustering obtained by Hoadley et al. (2014) of a pan-cancer dataset comprise measurements of five different 'omic types. The analysis of Hoadley *et al.* identified 11 major subtypes, showing that some cancer types were split into multiple clusters, while other (similar types of) cancers merged into the same cluster. The unsupervised analyses of the pan-cancer data presented in Chapters 3 and 4 confirmed the tumour subtyping defined by Hoadley *et al.*, with only a few discrepancies.

In addition to that, the outcome-guided integration of the pan-cancer data presented in Chapter 4 demonstrated that a more refined set of cancer subtypes may be discovered in this dataset. This demonstrates that outcome-guided KLIC could be applied in cancer subtyping and other precision medicine applications to uncover new subclusters.

Transcriptional module discovery

The goal of transcriptional module discovery is to find clusters of genes that are co-regulated and have the same biological function. The dataset that we used for this purpose contains gene expression and binding information for a large number of transcriptional regulators, in a species of yeast called *Saccaromyces cerevisiæ*. This dataset has already been analysed by others (Kirk et al., 2012; Savage et al., 2010).

In this thesis, the application of unsupervised KLIC to the yeast dataset served two main purposes. Firstly, showing the importance of selecting the correct clustering algorithm when doing consensus clustering and of checking that the resulting consensus matrix accurately reflects the amount of information present in the data (Chapter 3). Secondly, it demonstrated the advantage of using model-based clustering algorithms to construct the kernel matrices (Chapter 4).

6.2 FURTHER RESEARCH AREAS

Ideas for future research areas can be divided into two main topics: the extension of current methods, aimed at widening their applicability, and, for the unsupervised approaches, the improvement of current strategies for the evaluation and comparison of clustering results.

6.2.1 *Model extensions*

We can identify two types of model extensions that would make the proposed methods applicable to a wider range of datasets: the ability to handle missing values and to take as input different types of responses.

Handling missing values

The two medical applications presented in this thesis clearly highlighted the problem of missing data. Indeed, it is not trivial to measure and record large number of 'omic features for a set of individuals; this is in large part due to the fact that technical issues in the collection and processing of the 'omic data are not uncommon (see e.g. Troyanskaya et al., 2001).

We showed in Chapter 3 how missing values can be easily handled when using KLIC, assigning them no weight. Conversely, the two-step logistic regression methods developed in Chapter 2 and the DPMMs used in Chapter 4 cannot handle incomplete data. This led to a significant reduction of the sample size of each dataset and consequent loss of information.

As for the regression methods, a workaround is used in Seyres et al. (2020) to make sure that predictions are made for every individual included in the study, where models are fitted for any possible combination of 'omic layers. This way, all the available data are leveraged.

For DPMMs, instead, in order to handle missing values one could resort to the multiple imputation strategy suggested by Molitor et al. (2010) and Lunn et al. (2012, Chapter 9). The idea is to use the MCMC sampler to perform multiple imputation for the missing values.

Handling continuous and survival outcomes

The outcome-guided integration method of Chapter 4 can currently be used with categorical responses only. An extension to continuous responses should be straightforward, since the SVMs used in Chapter 4 can be easily adapted to the regression setting (see e.g. Bishop, 2006, Chapter 7). Similarly, the ability to use survival data to guide the kernel weighting would greatly increase the applicability of this approach, as many multi-omic studies use this as the response variable (see e.g. Zhao et al., 2015). This extension, however, would require developing and implementing new ways to compute the kernel weights.

On a related note, the two-step EN-type approaches of Chapter 2 have only been implemented and tested within the logistic regression framework. Implementing equivalent two-step approaches with linear regression and related models should be uncomplicated. A thorough assessment of the properties of these methods would then be required to ensure that they do not differ from those of the logistic regression framework considered here.

6.2.2 *Evaluation and comparison of clustering results*

Evaluating and comparing the output of clustering algorithms is a complex task. We divide the main questions arising from this thesis into three main points, which are all tightly linked.

Assessment of the similarity of two partitions

Following the introduction of the outcome-guided integration of multi-omic data in Chapter 4, it became apparent that the ARI is not a suitable measure to assess the similarity of two partitions, if one of the two is a more refined partition

than the other; empirical simulation studies could be used to prove this. Thus, a different metric need to be defined to assess the quality of a clustering in these situations.

Choice of the number of clusters

Using KLIC with consensus matrices requires choosing the number of clusters in each layer, in order to run CC, and in the final clustering. In Chapter 4, we solved the problem of choosing the number of clusters in the individual layers thanks to DPMMs. However, there remains the question of how to best pick the value of K for the global clustering. The silhouette can be a valuable tool, but it does not always give a clear answer, as shown by the outcome-guided integration of the pan-cancer data. For this reason, alternative strategies need to be devised.

Assessment of cluster quality

Finally, it is important to keep in mind that, before and after choosing the number of clusters, expert knowledge is required to assess the quality of clusters in real data applications.

6.3 CONCLUSIONS

In recent years, multi-omic analyses have allowed researchers to greatly improve their understanding of the flow of genetic information within a cell and how this affects different aspects of life. These analysis have driven the development of a large number of novel statistical and machine learning techniques. In this thesis, we have developed a set of tools that fall into this category, each tackling some of the challenges encountered in state-of-the-art approaches.

The work presented in this thesis has clearly demonstrated the importance of being able to monitor and/or determine how much each data layer contributes to the final output of every integrative multi-omic analysis, irrespectively of whether it is supervised or unsupervised. It has also highlighted the value of making use of a response variable associated to the outcome of interest in clustering applications, to guide the 'omic weighting process. The methodologies developed here allow to make a deliberate choice in terms of how much importance is given to each 'omic type, in different ways depending on whether a supervised, unsupervised, or outcome-guided approach is adopted. Additionally, balancing the sophistication of the techniques employed with the use of computationally efficient strategies (such as, for instance, always processing each data layer separately in the first step of each of the developed algorithms, in order to reduce the dimensionality of the data matrices involved) allowed us to keep the computational burden within reasonable bounds.

The diversity of high-throughput datasets is expected to further increase in the future. As well as increasing the potential for exciting discoveries, this will exacerbate the challenges of multi-omic integration discussed in this work (in terms of interpretability of the statistical models, computational cost, and so on) and bring new ones. In addition to the insights provided in this thesis, we hope that the techniques developed here (and corresponding implementations) will contribute to future scientific discoveries by other scientists and that they can be a starting point for further research in a number of directions.

A

APPENDIX TO CHAPTER 2

This appendix contains additional information on the work presented in Chapter 2. In Section A.1 we report the results of additional simulation studies for the penalised regression models considered in Section 2.3 and give more details about the choice of the parameter α for the first of our proposed approaches, where α is kept fixed. In Section A.2 we give more details about the biochemical parameters available for the CMS study. Section A.3 contains further details on the comparison between obese individuals and control donors presented in Section 2.5 (comparison 1) as well as the results obtained for the comparison between the lipodystrophy patients and the control donors (comparison 2).

A.1 ADDITIONAL SIMULATION STUDIES

We present here the results of three additional simulation settings. In the first one, only two penalised layers are combined. The other two are similar to the one presented in Section 2.3, with the number of penalised covariates increased from 2 to 10 and 100.

A.1.1 *Penalised covariates only*

Table A.1 contains the values of the parameters $P_1, P_2, P_N, P_1^r, P_2^r, \beta_1, \beta_2, \beta_N$ of the first additional simulation study. These are the same as those presented in Section 2.3, except that here $P_N = 0$ in all settings.

	P_1	P_2	P_N	P_1^r	P_2^r	β_1	β_2	β_N
Setting A	1000	1000	0	10	10	0.5	0.5	0.5
Setting B	100	1000	0	3	30	0.5	0.5	0.5
Setting C	100	1000	0	10	10	0.5	0.5	0.5
Setting D	100	1000	0	20	0	0.3	-	0.3
Setting E	20	1000	0	3	10	1	0.3	1
Setting F	20	1000	0	15	3	0.5	0.5	0.5

TABLE A.1: Values of $P_1, P_2, P_N, P_1^r, P_2^r, \beta_1, \beta_2, \beta_N$ used for the simulation study with penalised covariates only.

Since there are no covariates that are not penalised here, we also add the results obtained with a different univariate method. The idea is that for each variable a Mann-Whitney test is performed: if significant differences are observed between the two classes with confidence level 0.05, after adjusting for multiplicity using the Benjamini-Hochberg procedure, then that variable is selected. The variables selected in this way are then used to fit a ridge-penalised regression model.

In addition to that, for each method in each setting, we report the number of datasets for which each method is run successfully. When sIPF fails, this is due to the fact that all evaluations of the error surface have the same value. Results for the datasets affected by this problem are missing; these are only a small fraction of the total number of the generated datasets. Moreover, the two-step methods that fit a ridge regression model on the selected variables fail when only one variable is selected, since the `glmnet` implementation of logistic regression does not work with only one regressor.

The results are shown in Figures A.1 and A.2. In the case of diagonal covariance (Figure A.1), naïve EN has the highest MR, both within and out-of-sample. The number of variables selected by this method extremely low, so it is not surprising to observe higher precision and lower recall compared to the other methods. The low number of selections is due to the fact that naïve-EN, like sIPF-EN, is able to identify sets of features from different layers that together are predictive of outcome status. As in the simulation setting presented in Chapter 2, the two-step EN-type methods have very low values of within-sample MR, but similar values of out-of-sample MR to sIPF-EN in the first four settings. Interestingly, in settings E and F, which are highly unbalanced, sIPF-EN greatly outperforms the other two algorithms both in terms of out-of-sample MR and precision.

In the non-diagonal covariance case, similar outcomes are observed. The main difference is that naïve-EN has lower MR and higher recall in this setting. In both cases, the two univariate methods behave similarly, selecting fewer variable than the other methods (as expected) and achieving a low MR only in Setting E.

A.1.2 *Higher number of non-penalised covariates*

We repeat the experiments presented in Chapter 2, replacing the value of P_N to 10 (Figures A.3 and A.4) and 100 (Figures A.5 and A.6).

The patterns are similar to those observed in Chapter 2, with the number of selected variables and values of the recall becoming more and more different between the joint models (naïve-EN and sIPF-EN) and the two-step approaches as P_N increases. The univariate method cannot be run in the setting where $P_N = 100$, since the number of predictors is higher than the number of statistical units in the screening step, and a linear regression model cannot be fit.

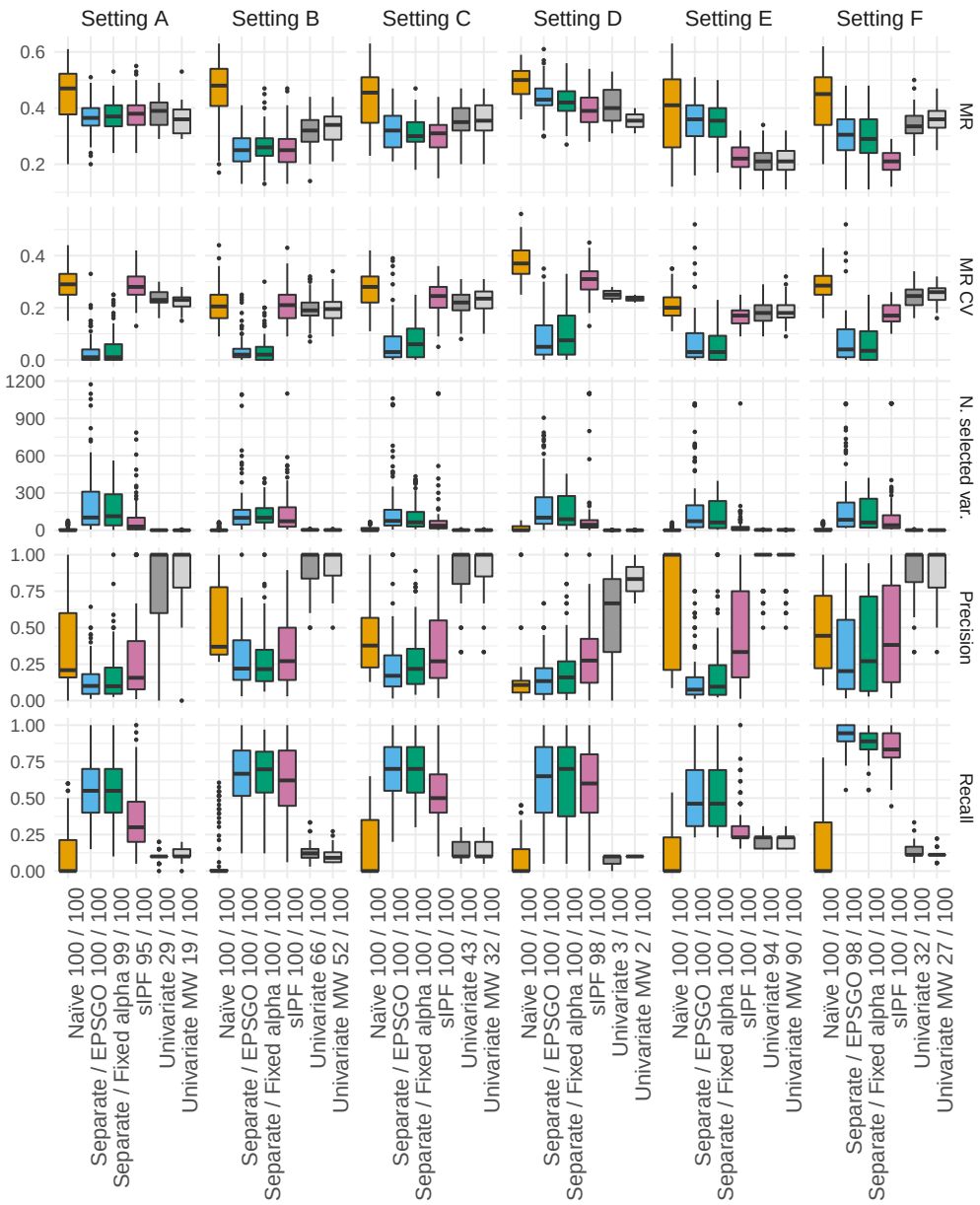


FIGURE A.1: Simulation study comparing different variants of elastic-net for multi-omic data. The covariance matrix used here is the diagonal matrix Σ_0. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate.

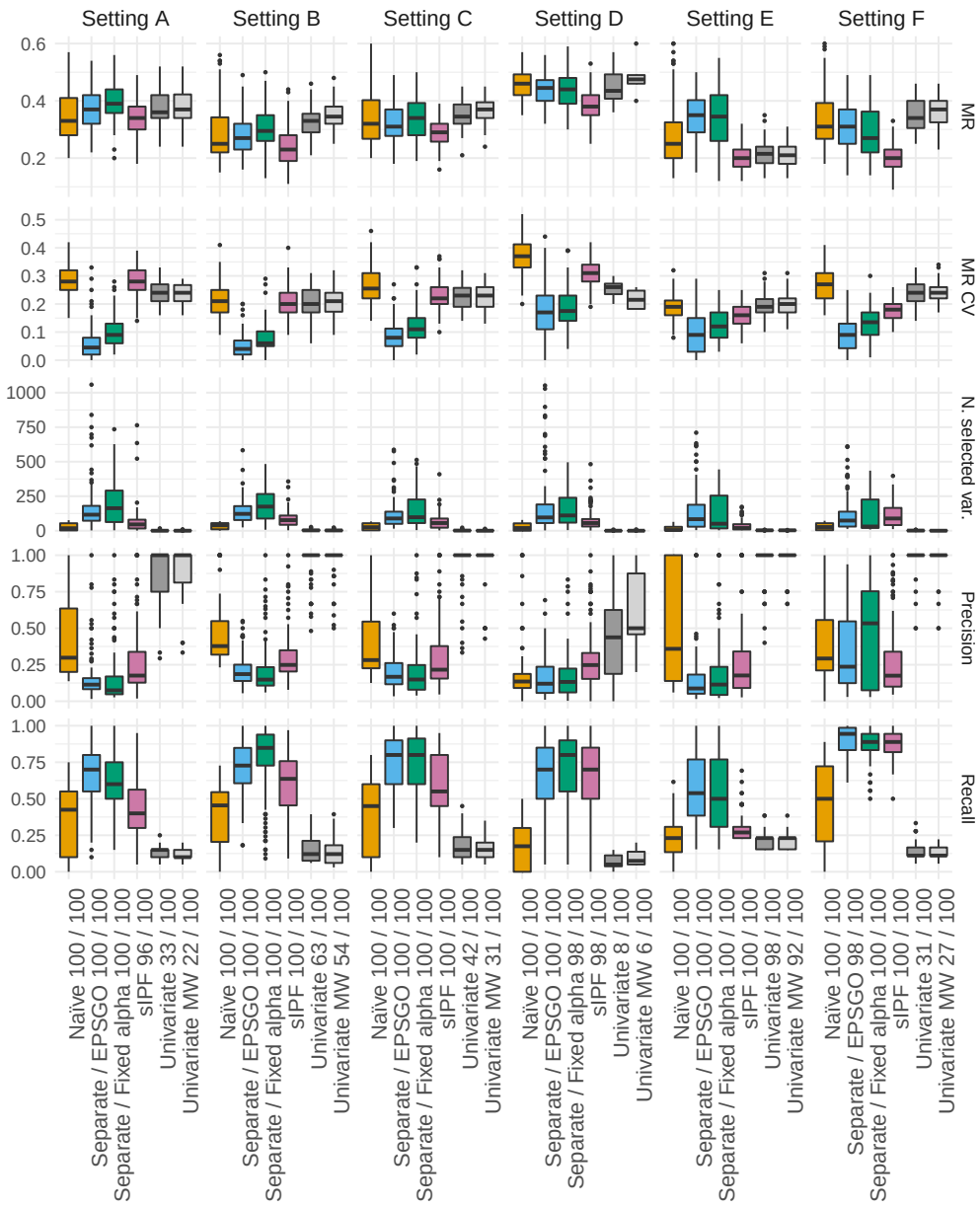


FIGURE A.2: Simulation study comparing different variants of elastic-net for multi-omic data. The covariance matrix used here is the block matrix Σ_1. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate.

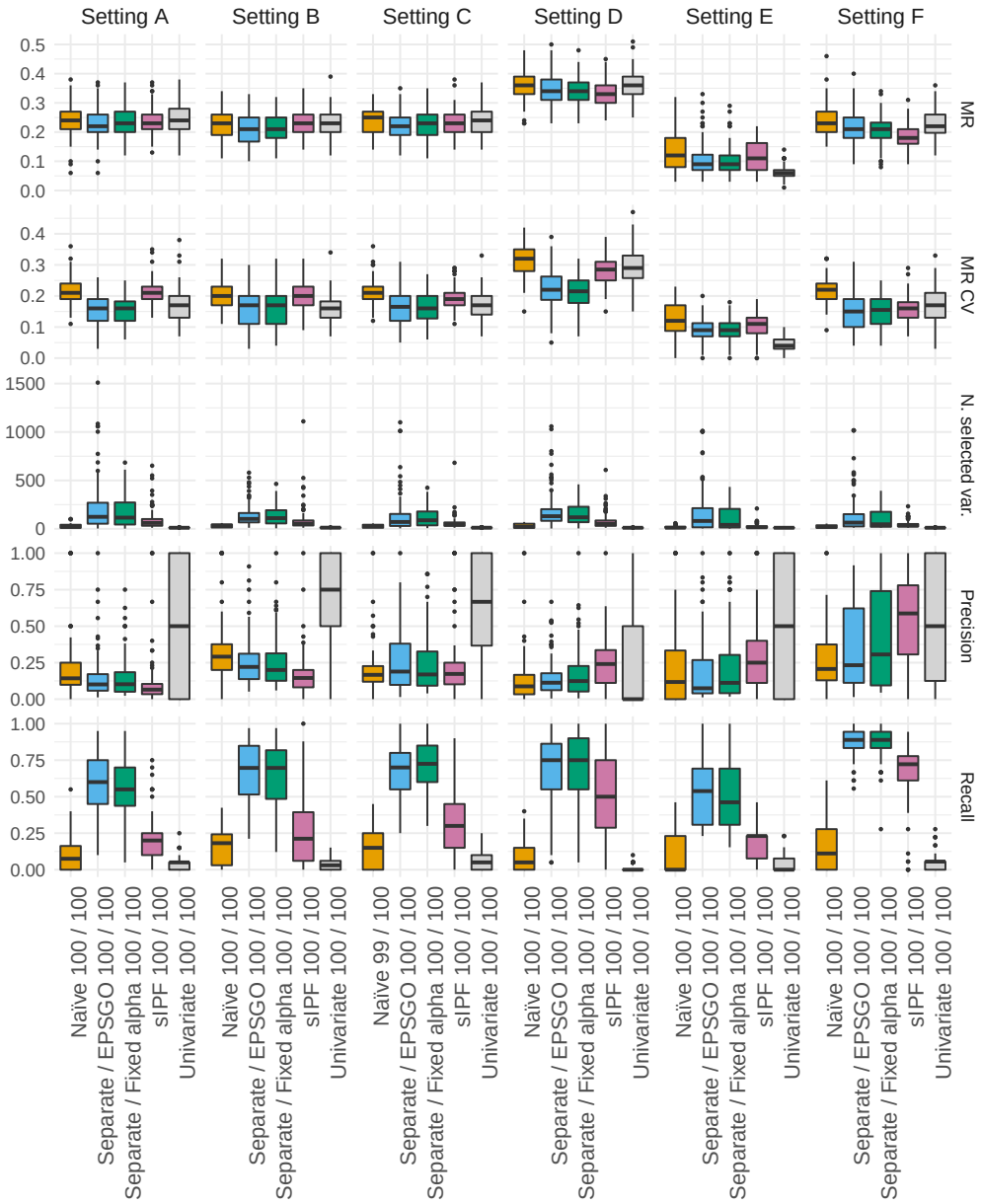


FIGURE A.3: Simulation study comparing different variants of elastic-net for multi-omic data. The covariance matrix used here is the diagonal matrix Σ_0. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

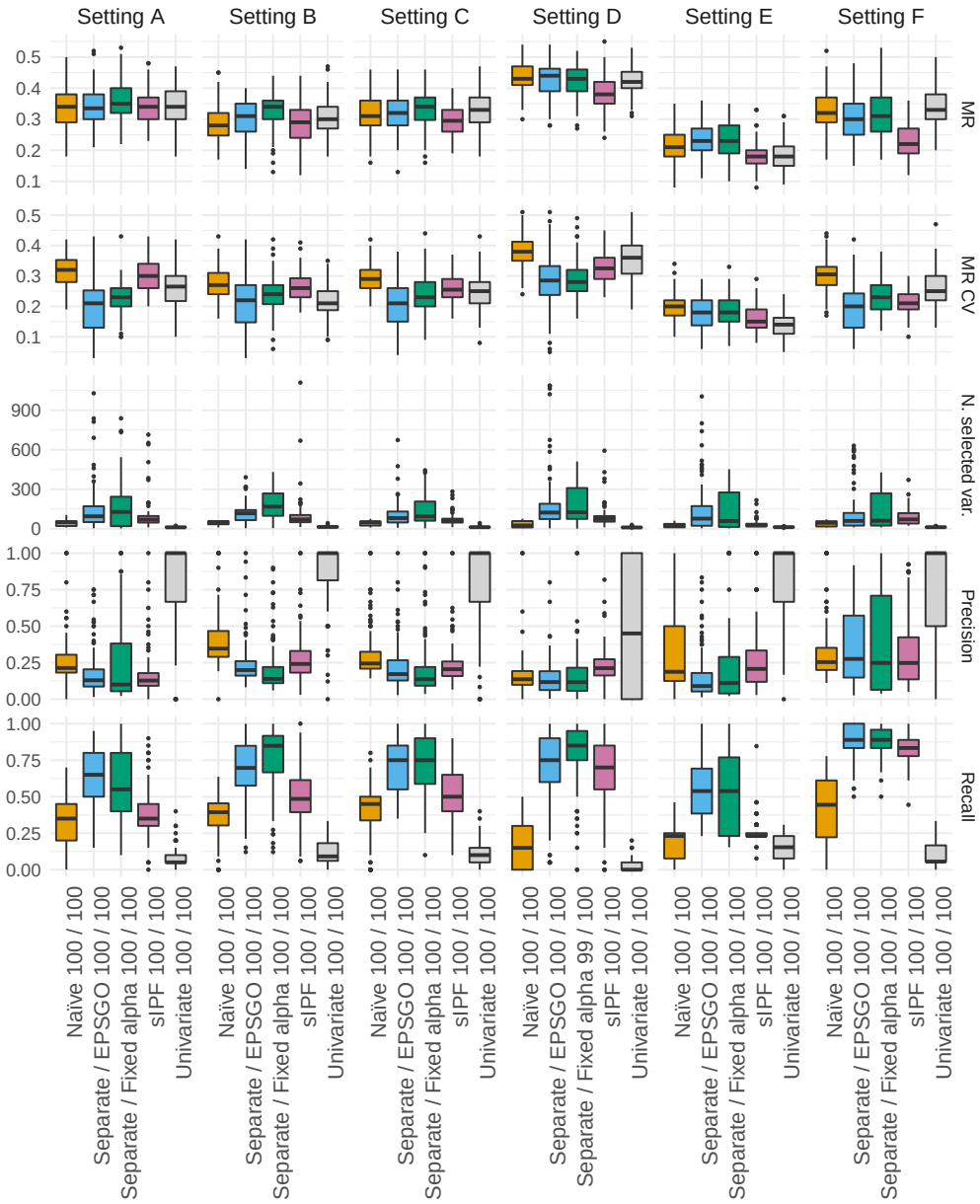


FIGURE A.4: Simulation study comparing different variants of elastic-net for multi-omic data. The covariance matrix used here is the block matrix Σ_1. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

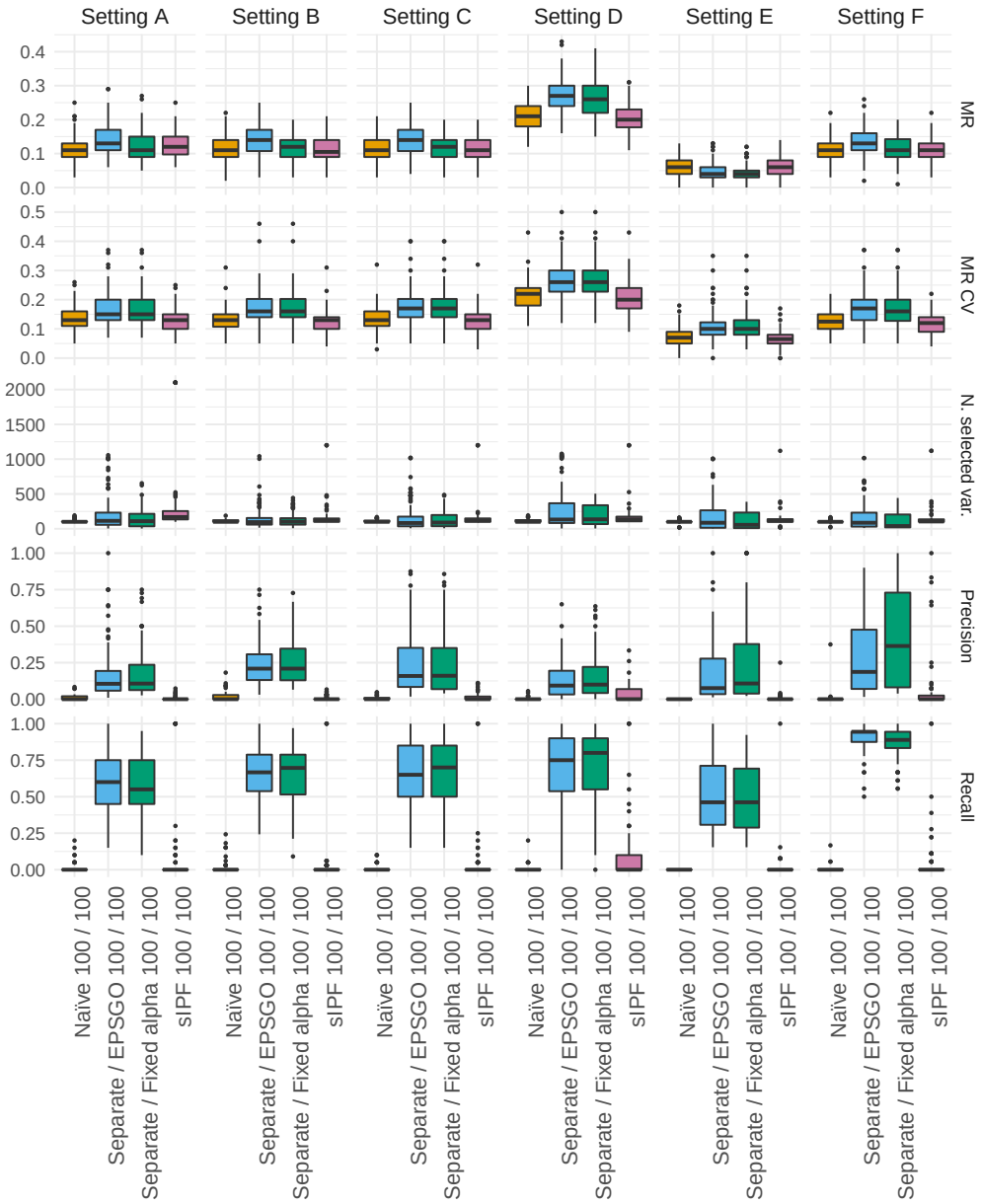


FIGURE A.5: Simulation study comparing different variants of elastic-net for multi-omic data. The covariance matrix used here is the diagonal matrix Σ_0. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

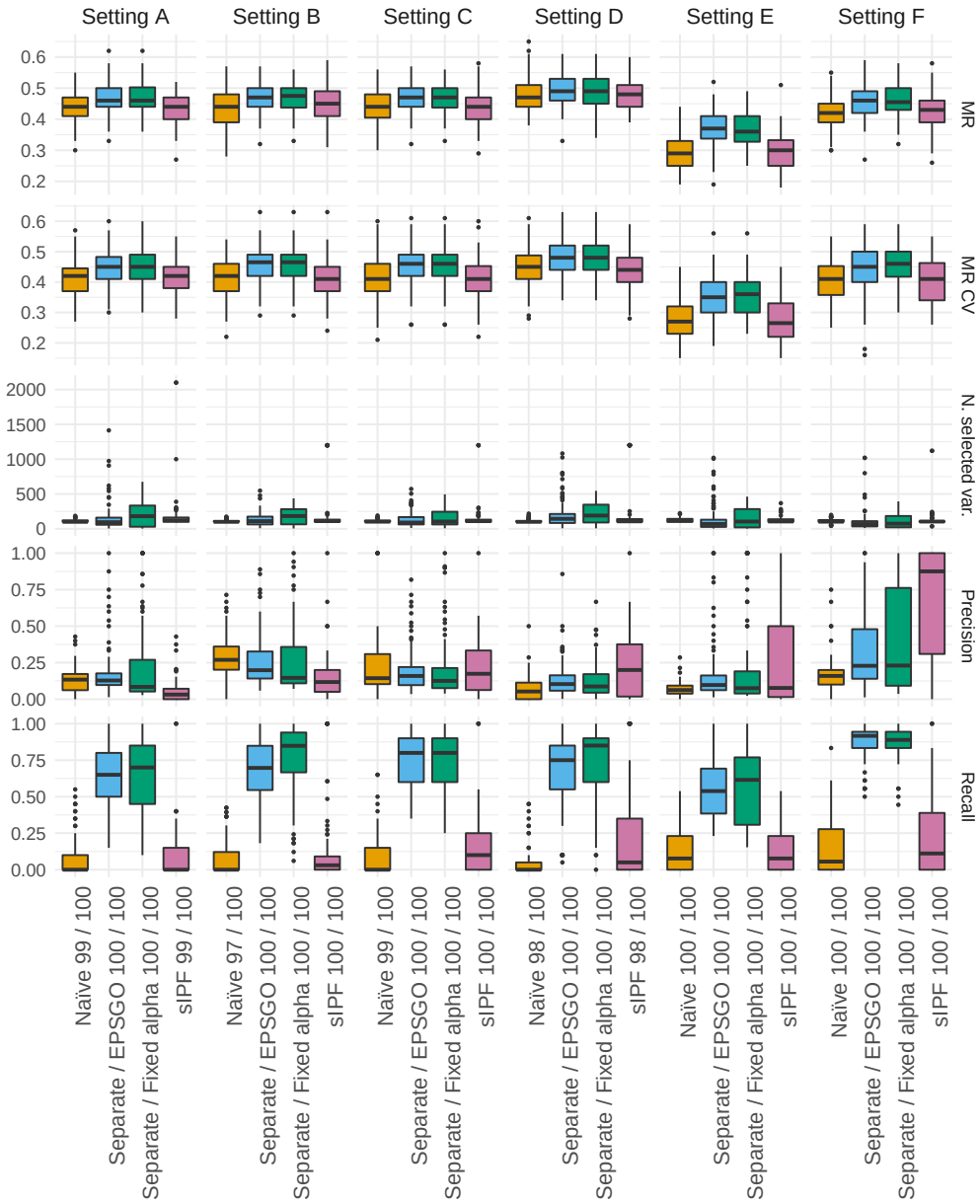


FIGURE A.6: Simulation study comparing different variants of elastic-net for multi-omic data. The covariance matrix used here is the block matrix Σ_1. MR is the out-of-sample misclassification rate, MR CV the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

A.1.3 *Choice of α*

We compare different values of α for the two-step approach proposed in Chapter 2, where α is kept fixed. We consider $\alpha = 0.1, 0.5$ and 1 and all the simulations settings compared so far: only two penalised layers (Figures A.7 and A.8), two non-penalised covariates (Figures A.9 and A.10), few non-penalised covariates (Figures A.11 and A.12), and high number of non-penalised covariates (Figures A.13 and A.14).

The number of selected variables, precision and recall are as expected: the number of selected variables and the recall decrease while the precision increases as as the value of α is increased. The MR, however, does not follow a clear pattern. Table A.2 shows how the out-of-sample MR varies in each simulation setting for increasing values of α.

$\boldsymbol{P_N}$	0	0	2	2	10	10	100	100
Covariance	Σ_0	Σ_1	Σ_0	Σ_1	Σ_0	Σ_1	Σ_0	Σ_1
Setting A	$=$	$\downarrow$	$=$	$\downarrow$	$\downarrow$	$\uparrow$	$\uparrow$	$\downarrow$
Setting B	$=$	$\downarrow$	$=$	$\downarrow$	$\downarrow$	$\downarrow$	$=$	$=$
Setting C	$=$	$\downarrow$	$=$	$\downarrow$	?	$\downarrow$	$=$	$=$
Setting D	$=$	$\downarrow$	$\uparrow$	$=$	$=$	$=$	$=$	$=$
Setting E	$\downarrow$	$\downarrow$	$=$	$=$	$\uparrow$	$=$	$=$	$\downarrow$
Setting F	$\downarrow$	?	$=$	$=$	$\uparrow$	$\downarrow$	$\uparrow$	$\downarrow$

TABLE A.2: Variation in the median MR of the separate EN approach with fixed α when the parameter α is increased.

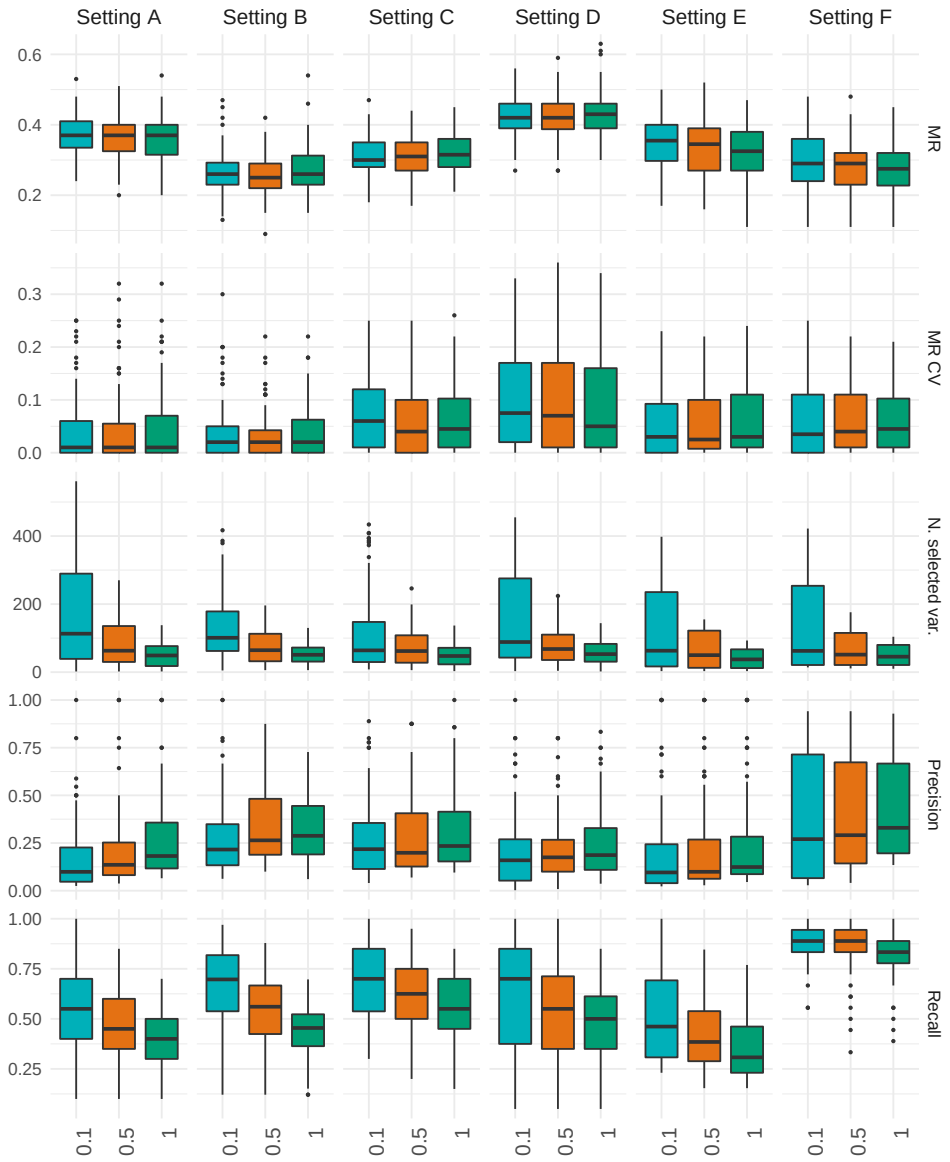


FIGURE A.7: Simulation study comparing different values of α. The covariance matrix used here is the diagonal matrix Σ_0 and $P_N = 0$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate.

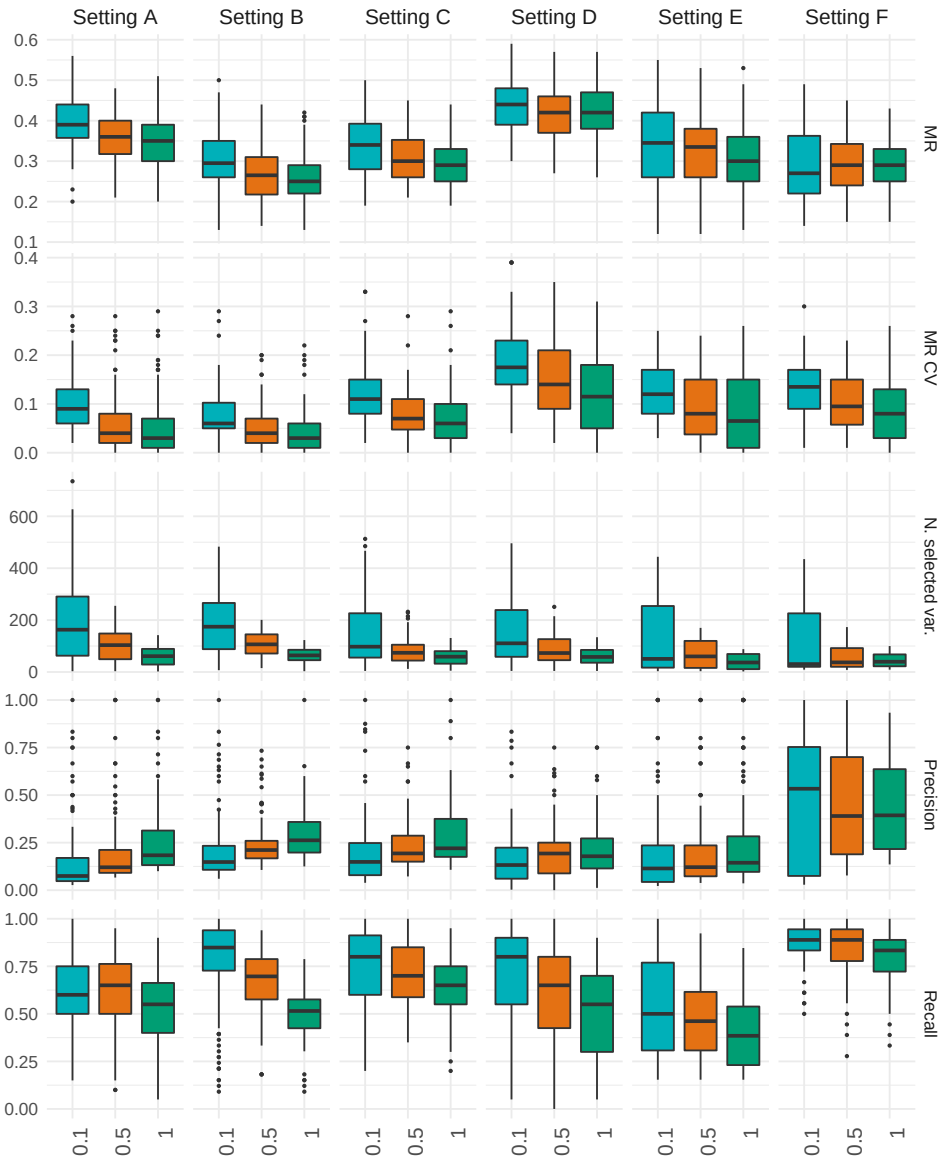


FIGURE A.8: Simulation study comparing different values of α. The covariance matrix used here is the block matrix Σ_1 and $P_N = 0$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate.

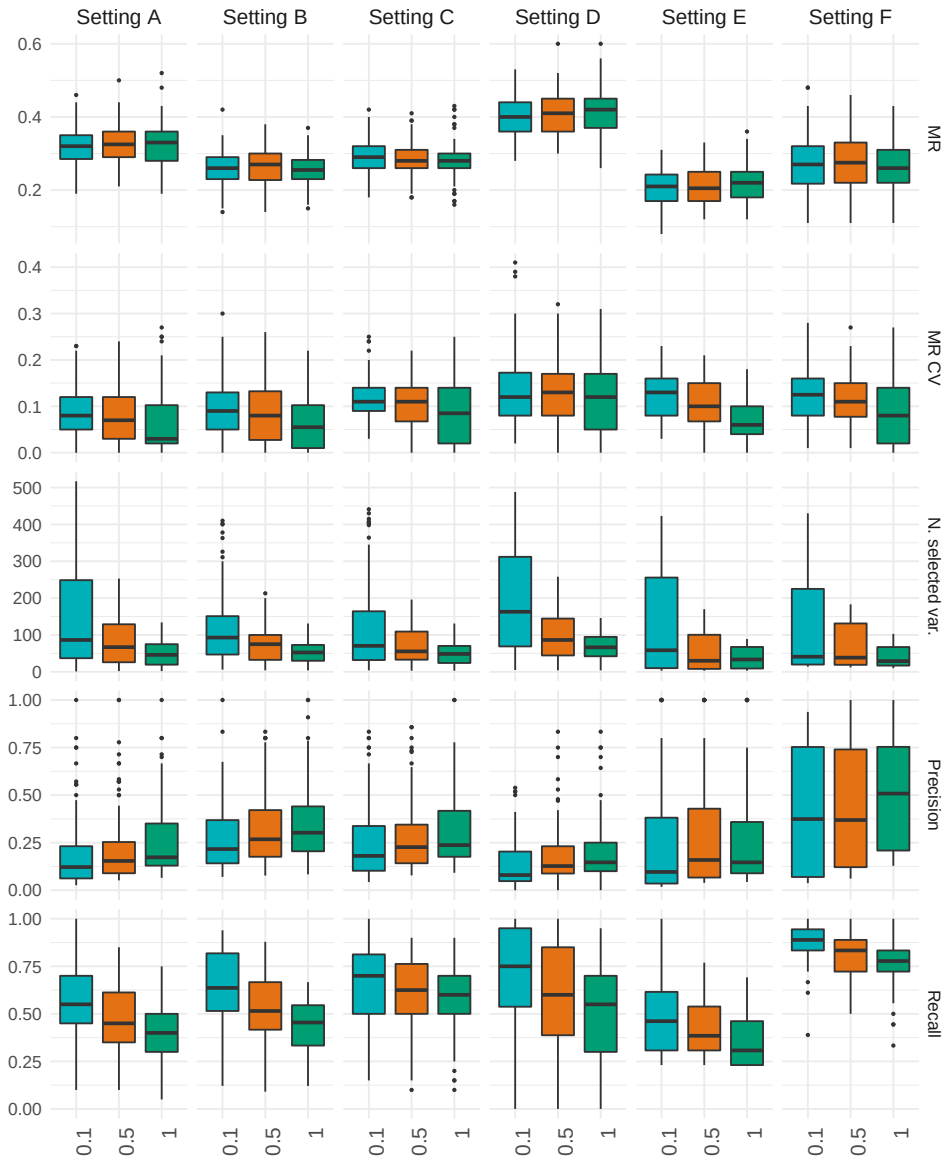


FIGURE A.9: Simulation study comparing different values of α. The covariance matrix used here is the diagonal matrix Σ_0 and $P_N = 2$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate.

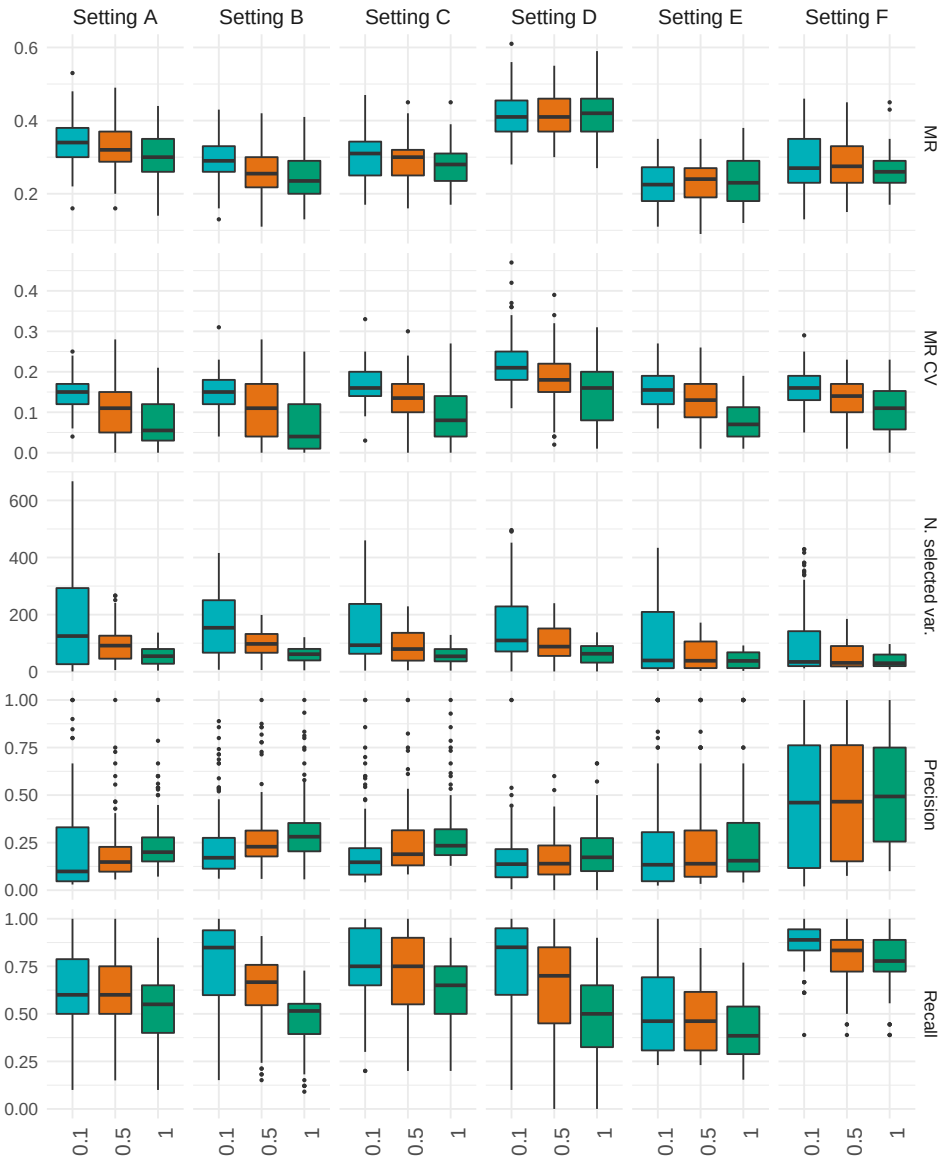


FIGURE A.10: Simulation study comparing different values of α. The covariance matrix used here is the block matrix Σ_1 and $P_N = 2$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate.

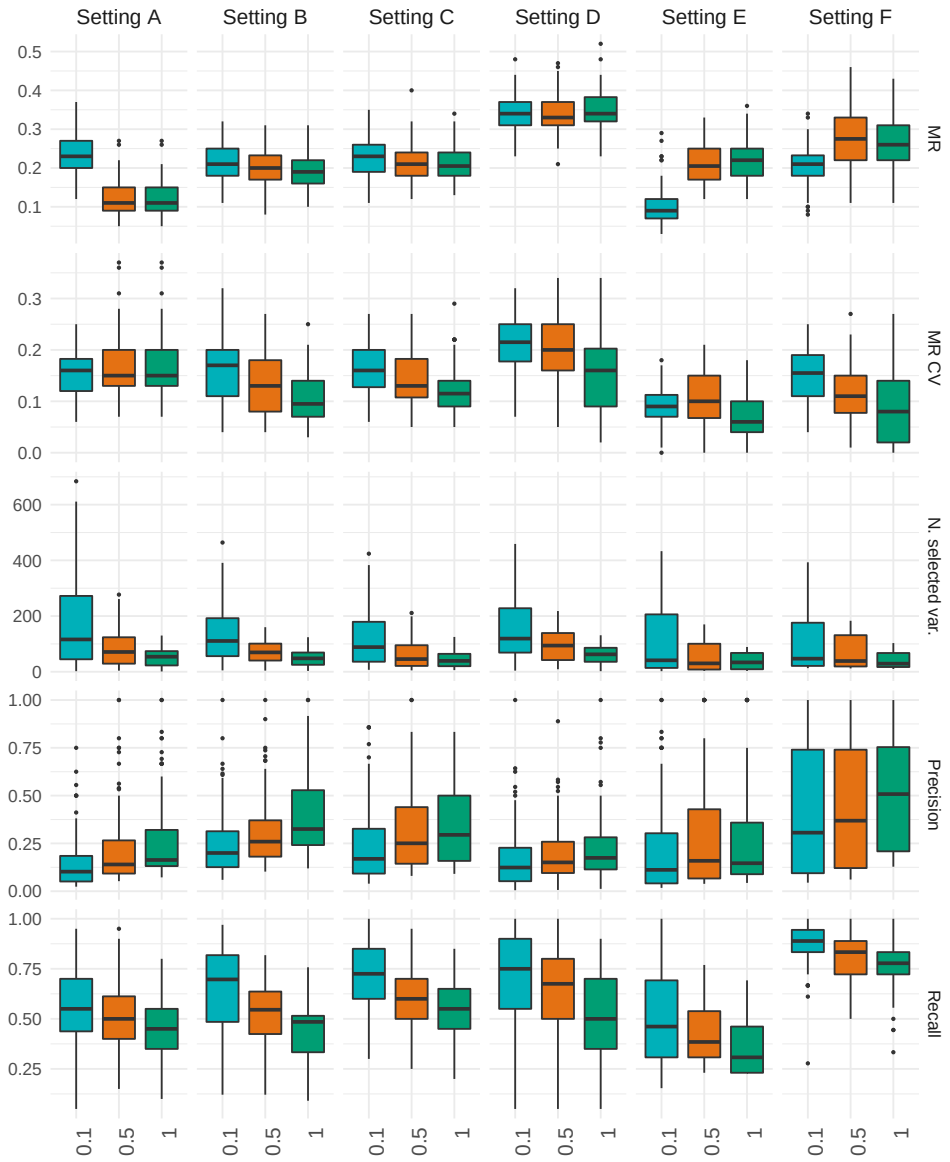


FIGURE A.11: Simulation study comparing different values of α. The covariance matrix used here is the diagonal matrix Σ_0 and $P_N = 10$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate.

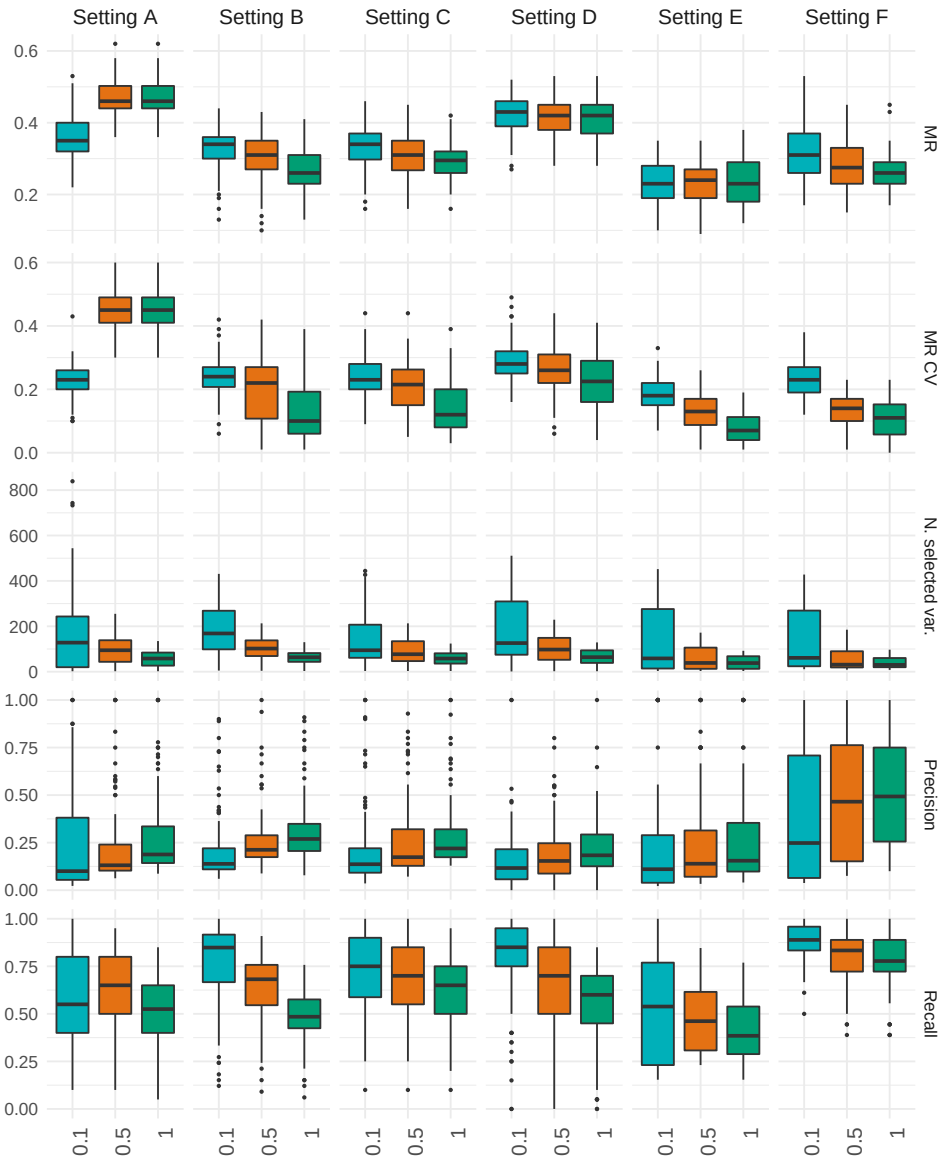


FIGURE A.12: Simulation study comparing different values of α. The covariance matrix used here is the block matrix Σ_1 and $P_N = 10$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate.

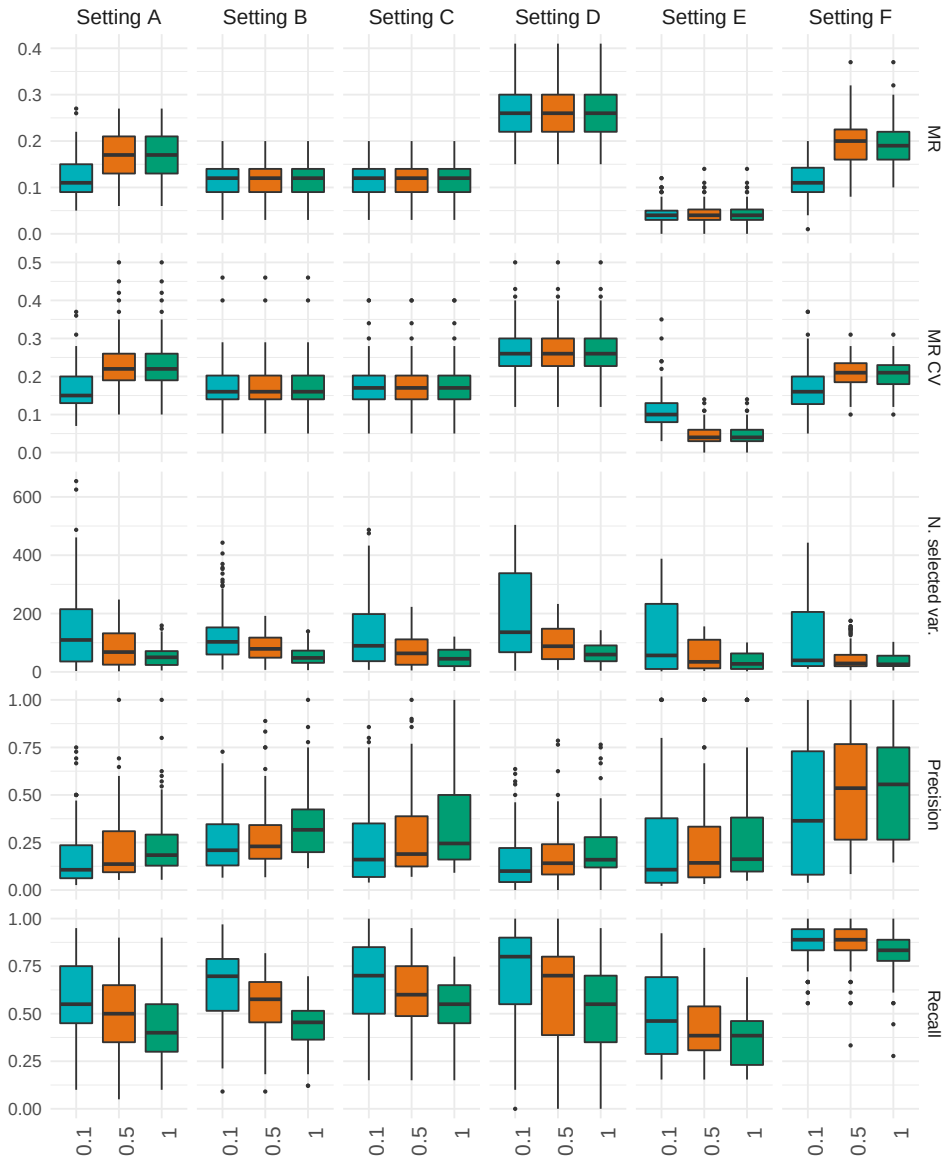


FIGURE A.13: Simulation study comparing different values of α. The covariance matrix used here is the diagonal matrix Σ_0 and $P_N = 100$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

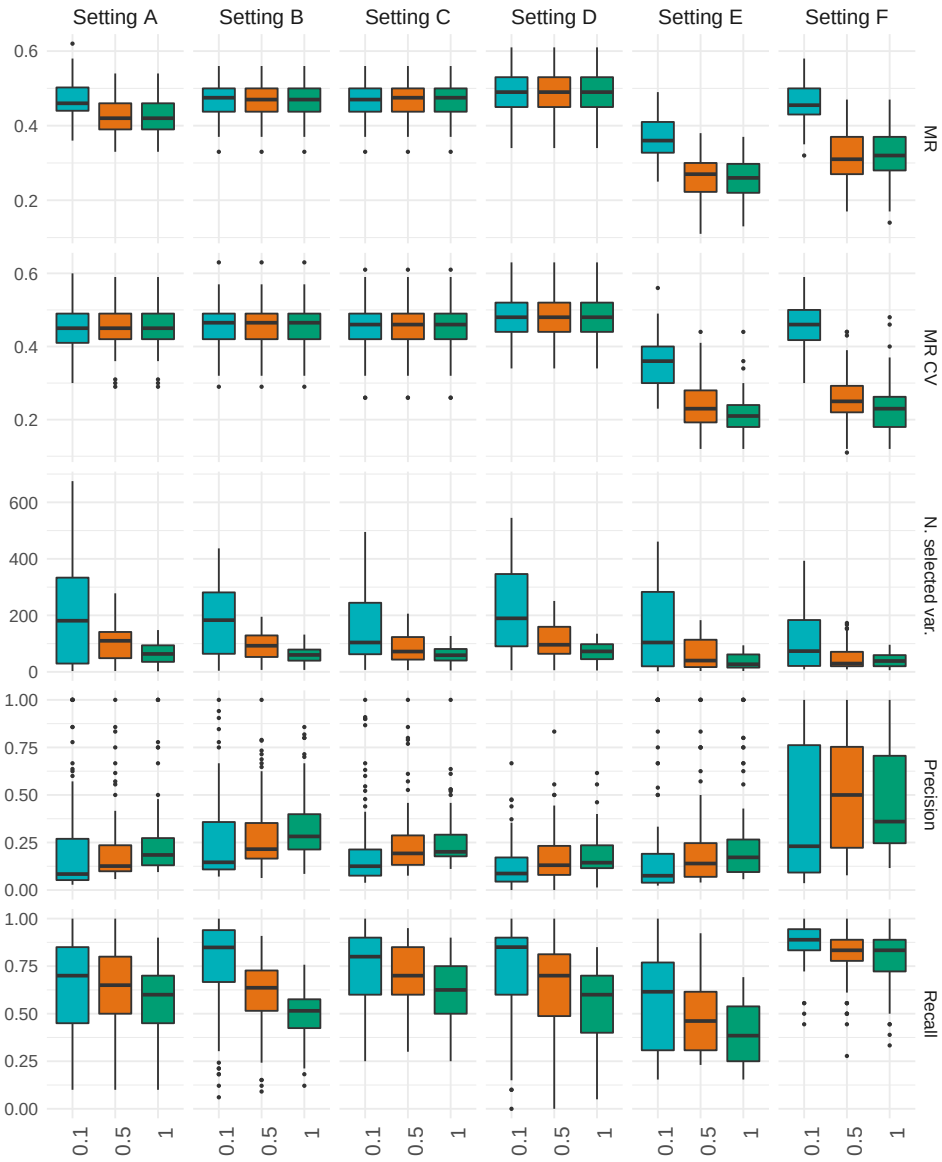


FIGURE A.14: Simulation study comparing different values of α. The covariance matrix used here is the block matrix Σ_1 and $P_N = 100$. MR is the out-of-sample misclassification rate, MR CV is the within-sample misclassification rate. The non-penalised covariates are not included when computing precision and recall.

A.2 DESCRIPTION OF EACH BIOCHEMICAL PARAMETER

The biochemical parameters measured for the CMS study are the following:

- *Alanine amino-transferase (ALT)*, an enzyme found primarily in the liver and kidney. It is considered to be a biomarker for liver health. Elevated levels of ALT often suggest the existence of liver disease (Lala and Minter, 2018).
- *Aspartate amino-transferase (AST)*, an enzyme produced mainly by the liver, but is also found in small quantities in other organs. As for ALT, high levels of AST are usually associated to liver damage or other medical problems (Lala and Minter, 2018).
- *Glucose*, a sugar that is critical for the functioning of a number of tissues of the human body. In healthy individuals, blood glucose level is tightly regulated and remains constant. High blood glucose levels are usually due to diabetes (Wasserman, 2009).
- *Low-density lipoprotein cholesterol (LDL-C)*, also known as "bad cholesterol". Elevated levels of LDL-C correlate with the extent and progress of atherosclerosis. In healthy individuals this constitutes around 60%-70% of blood cholesterol (Hamilton, 1997).
- *High-density lipoprotein cholesterol (HDL-C)*, also known as "good cholesterol". In healthy individuals this constitutes around 20%-30% of blood cholesterol. High levels of HDL-C are correlated with better cardiovascular health (Sirtori, 2006).
- *Triglycerides (TG)*, the main constituent of body fat in humans. The triglycerides present in the blood enable the transfer of adipose fat and blood glucose from the liver and vice versa. High levels of triglycerides in the bloodstream have been linked to atherosclerosis, heart disease, and stroke (Nordestgaard et al., 2007; Sarwar et al., 2007; Talayero and Sacks, 2011).
- *Total cholesterol*, which includes LDL-C, HDL-C, and tryglicerides.
- *High sensitivity C-reactive protein (hsCRP)*. Increased levels of C-reactive protein in the blood indicate the presence of an inflammation. Since atherosclerosis is believed to be a long-term mild inflammatory process, the high sensitivity C-reactive can detect slightly increased levels of C-reactive protein and, for this reason, it is used to predict a person's risk of myocardial infarction and stroke (Ridker, 2001).
- *Insulin*, a hormone produced by the pancreas in response to carbohydrates consumed in the diet. It promotes the absorption of carbohydrates from the blood into liver, fat and skeletal muscle cells. Obesity has been shown to

be associated with an increased risk of developing insulin resistance (Kahn, Hull, and Utzschneider, 2006).

- *Free fatty acid (FFA)* are fatty acids circulating in the plasma. Increasing levels of obesity are often associated with increasing levels of circulating FFA (Gordon, 1960; Reaven et al., 1988).
- *Leptin*, a hormone that helps regulate the balance between food intake and energy expenditure by inhibiting hunger. Obese individuals are known to be less sensitive to leptin (de Gusmao Correia and Haynes, 2004).
- *Adiponectin*, a hormone produced in adipose tissue that is involved in regulating glucose levels and fatty acid breakdown. Low levels of adiponectin are asociated with diabetes and cardiovascular disease (Oh, Ciaraldi, and Henry, 2007).

Furthermore, the following quantities were computed:

- *Leptin to adiponectin ratio (LAR)*, a marker of atherosclerosis susceptibility (Kieć-Klimczak, Malczewska-Malec, and Huszno, 2008).
- *Homeostasis model assessment of insulin resistance (HOMA-IR) score*, a way of quantifying insuline resistance, a pathological condition in which cells fail to respond normally to the hormone insulin. In states of insulin resistance, the same amount of insulin does not have the same effect on glucose transport and blood sugar levels (Katsuki et al., 2001).
- *Adipose tissue insuline resistance (ADIPO-IR) score*, a measure of adipose tissue insulin sensitivity. Low insulin sensitivity is associated with type 2 diabetes (Søndergaard et al., 2017).

A.3 ADDITIONAL DATA ANALYSIS

A.3.1 *Obese individuals versus control donors*

We report here some additional results of the first comparison, that were omitted from Chapter 2 for the sake of brevity.

Comparison of multivariate and univariate variable selection

In Section 2.6 is reported a Venn diagram representing the intersection between the number of variables selected by the elastic-net (considering both the median and maximal sets of variables over multiple runs of 10-fold CV) and those selected via univariate testing for the lipidomics layer. Table A.3 contains the same information for all layers.

Signature validation

In order to validate our multivariate signatures of CMS, we check whether there is an association between our selected variables and the anthropometric and biochemical parameters. To do so, we fit the following regression model, that adjusts for age and sex, for each pair of anthropometric/biochemical parameter and 'omic measurement:

$$\text{parameter}_i = \beta_0 + \beta_{\text{age}} \times \text{age}_i + \beta_{\text{female}} \times \mathbb{1}(\text{female}_i) + \beta_{\text{'omic}} \times \text{'omic}_i + \epsilon,$$

where $i = 1, \ldots, N$ and $\mathbb{1}$ is the indicator function.

Figures A.17 and A.19 show the coefficients of the selected metabolites and lipids in each of these regressions. Those marked with a star correspond to those for which the null hypothesis is rejected for the test $H_0 : \beta_{\text{'omic}} = 0$ versus $H_1 : \beta_{\text{'omic}} \neq 0$ at significance level 0.01. We control the FDR by adjusting the p-values using the Benjamini-Hochberg procedure.

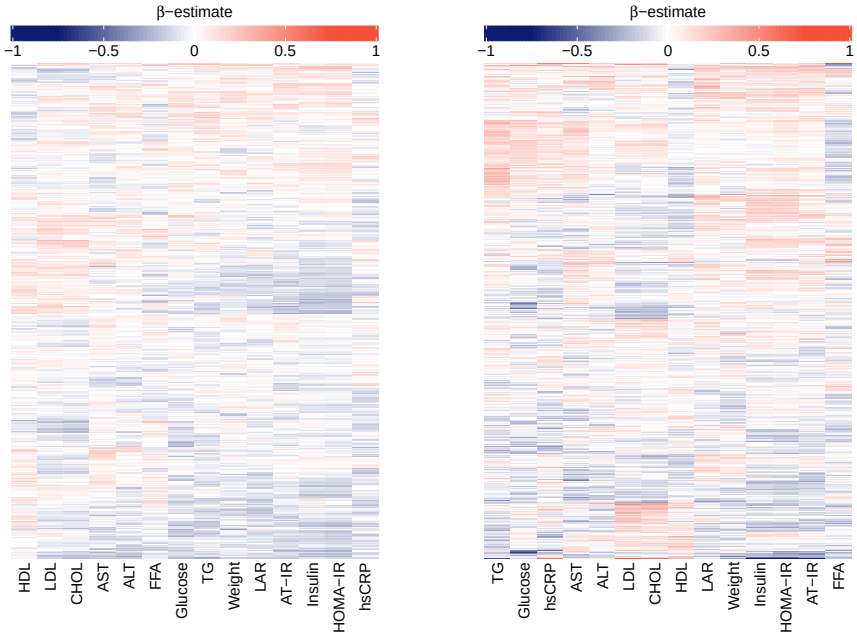
The coefficients of regression model of Equation (2.3) for the ChIP-seq, RNA-seq, and methylation data are shown in Figures A.15, A.16, and A.17. In these datasets, the number of tests that are significant at level 0.01 after correcting for multiple testing using the Benjamini-Hochberg procedure is low. The ChIP-seq and methylation variables have no significant associations, while the RNA-seq only a few.

	Max	Mode	Univ.	Max∩Mode	Max∩Univ.	Mode∩Univ.	All
ChIP-seq / Monocytes	428	350	0	350	0	0	0
ChIP-seq / Neutrophils	611	611	0	611	0	0	0
RNA-seq / Monocytes	425	425	0	425	0	0	0
RNA-seq / Neutrophils	592	588	0	588	0	0	0
Methylation / Monocytes	106	45	0	45	0	0	0
Methylation / Neutrophils	25	6	0	6	0	0	0
Metabolites	60	10	0	10	0	0	0
Lipids	62	61	14	61	0	14	14

TABLE A.3: Intersections between the sets variables selected with the multivariate approach with the maximal and modal set of variables and those selected via univariate testing, model trained on the lipodystrophy patients and control donors.

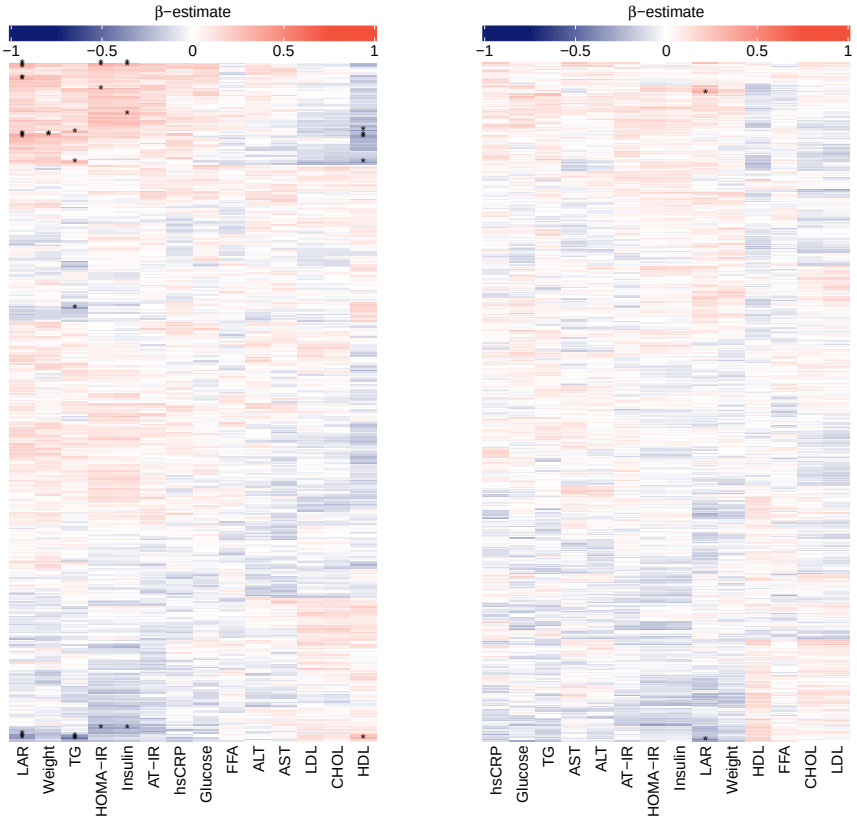
	Max	Mode	Univ.	Max∩Mode	Max∩Univ.	Mode∩Univ.	All
ChIP-seq / Monocytes	582	577	0	577	0	0	0
ChIP-seq / Neutrophils	565	440	0	440	0	0	0
RNA-seq / Monocytes	455	455	0	455	0	0	0
RNA-seq / Neutrophils	630	623	0	623	0	0	0
Methylation / Monocytes	128	47	0	47	0	0	0
Methylation / Neutrophils	71	0	0	0	0	0	0
Metabolites	232	131	0	131	0	0	0
Lipids	73	68	58	68	40	40	40

TABLE A.4: Intersections between the sets variables selected with the multivariate approach with the maximal and modal set of variables and those selected via univariate testing, model trained on the lipodystrophy patients and control donors.



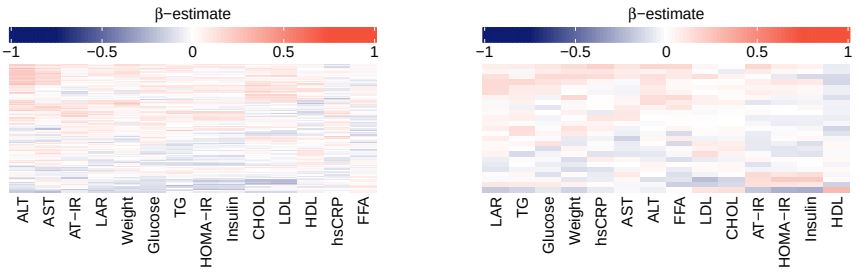
(A) Monocytes. (B) Neutrophils.

FIGURE A.15: Association of the selected ChIP-seq peaks with the anthropometric and biochemical parameters. Cells marked with a star represent associations that are statistically significant with a confidence level of 0.01 after correcting for multiple testing using the Benjamini-Hochberg procedure.



(A) Monocytes. (B) Neutrophils.

FIGURE A.16: Association of the selected RNA-seq transcripts with the anthropometric and biochemical parameters. Cells marked with a star represent associations that are statistically significant with a confidence level of 0.01 after correcting for multiple testing using the Benjamini-Hochberg procedure.



(A) Monocytes. (B) Neutrophils.

FIGURE A.17: Association of the selected methylation sites with the anthropometric and biochemical parameters. Cells marked with a star represent associations that are statistically significant with a confidence level of 0.01 after correcting for multiple testing using the Benjamini-Hochberg procedure.

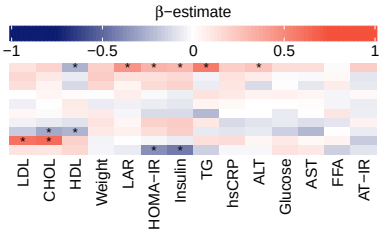


FIGURE A.18: Association of the selected metabolites with the anthropometric and biochemical parameters. Cells marked with a star represent associations that are statistically significant with a confidence level of 0.01 after correcting for multiple testing using the Benjamini-Hochberg procedure.

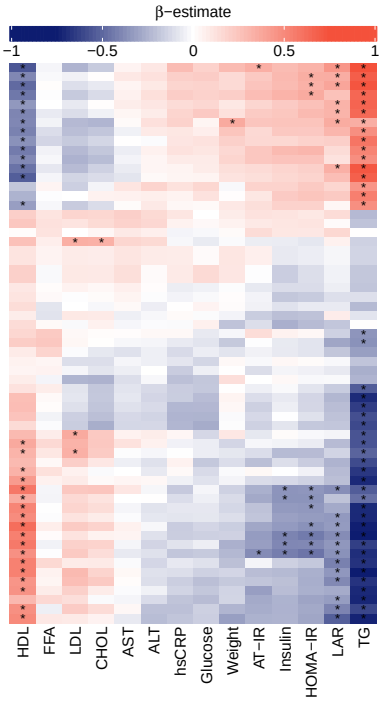


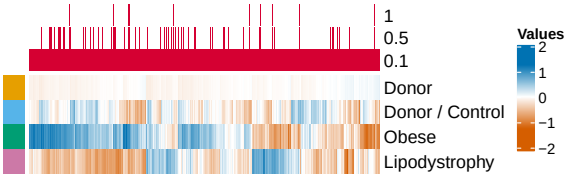
FIGURE A.19: Association of the selected lipids with the anthropometric and biochemical parameters. Cells marked with a star represent associations that are statistically significant with a confidence level of 0.01 after correcting for multiple testing using the Benjamini-Hochberg procedure.

A.3.2 *Lipodystrophy patients versus control donors*

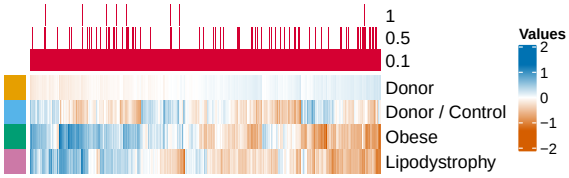
Figure A.20 shows the average values of the selected variables for comparison 2 (lipodystrophy patients versus control donors). Interestingly, lipodystrophy and obese individuals have similar average values in neutrophils for ChIP-seq and RNA-seq data, but the same is not true for the other datasets.

In Figure A.21 are reported the probabilities of being a *case*, i.e. lipodystrophy patient, for each individual and each layer, as well as those given by a ridge regression on the clinical covariates. The probability given by the full model including all the selected variables is also reported. Moreover, the rankings of each person by probability of being a case for each layer and the set of clinical covariates are reported, together with the final, average ranking. The results are comparable to those obtained when using the obese individuals for the training set, where all patients have high probabilities and rankings.

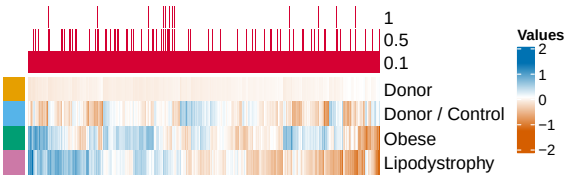
Table A.4 contains the number of variables selected by the multivariate and univariate analyses and their intersections.



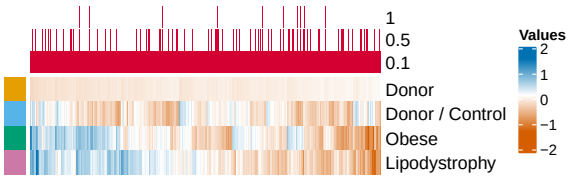
(A) ChIP-seq, neutrophils.



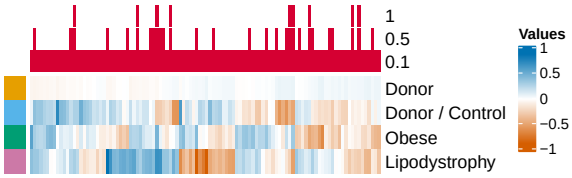
(B) ChIP-seq, neutrophils.



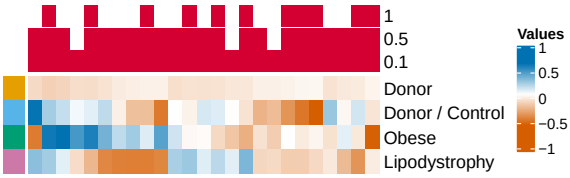
(C) RNA-seq, monocytes.



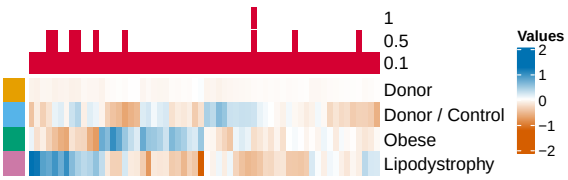
(D) RNA-seq, neutrophils.



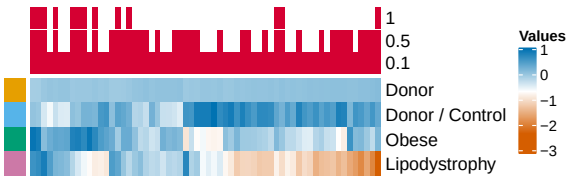
(E) Methylation, monocytes.



(F) Methylation, neutrophils.

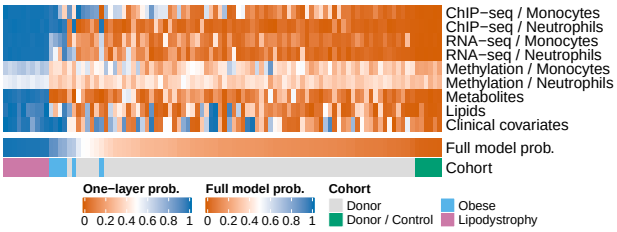


(G) Metabolites.

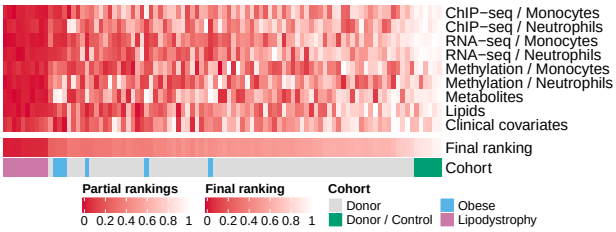


(H) Lipids.

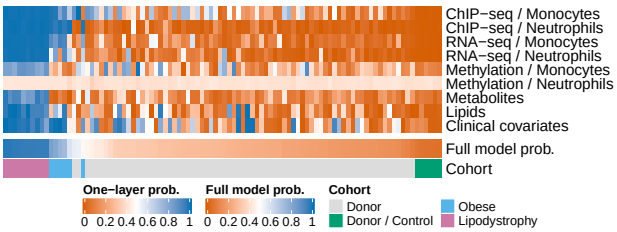
FIGURE A.20: Signature identification. Penalised logistic regression model trained on the lipodystrophy patients and control donors. Average values of the selected variables for each group of people. Each column corresponds to one of the selected variables.



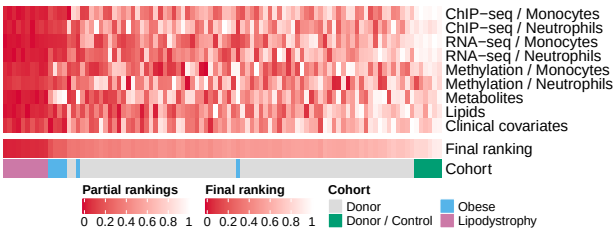
(A) Probability of being a *case*, $\alpha = 0.1$.



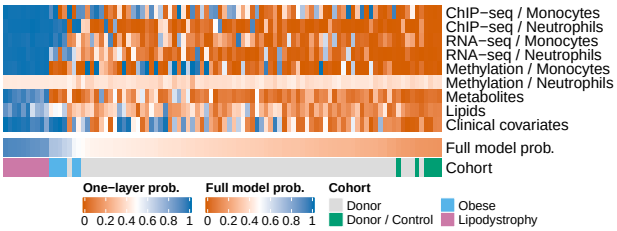
(B) Combined rankings, $\alpha = 0.1$.



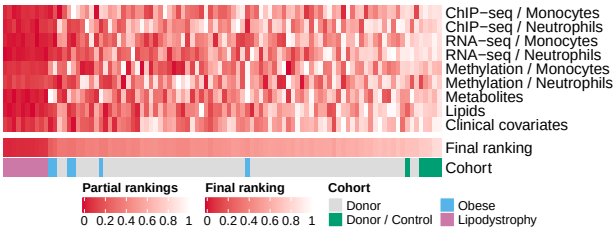
(C) Probability of being a *case*, $\alpha = 0.5$.



(D) Combined rankings, $\alpha = 0.5$.



(E) Probability of being a *case*, $\alpha = 1$.



(F) Combined rankings, $\alpha = 1$.

FIGURE A.21: Probabilities of belonging to the class of lipodystrophy patients and ranking of each person according to those probabilities. Both quantities are shown on each dataset separately and considering all the data types jointly. The model is trained on the lipodystrophy patients and control donors. Each column corresponds to one of the individuals who have no missing data, each row corresponds to one of the layers. The columns are sorted by probability of being a case in (A), (C), and (E) and final ranking in (B), (D), and (F). All rankings are divided by the total number of observations.

B

APPENDIX TO CHAPTER 3

This appendix contains additional information on the work presented in Chapter 3. In Section B.1 additional figures and results for the simulation studies of Section 3.5 are reported. In Sections B.2 and B.3 are reported additional figures and analyses for the two biological applications of Sections 3.6 and 3.7 respectively.

B.1 SIMULATION STUDY

In Section B.1.1 we explain how the RBF parameter was tuned for the simulation studies. Additional settings for the comparison between KLIC, COCA and other clustering algorithms are presented in Section B.1.2.

B.1.1 *Choice of the parameter of the radial basis function kernels*

In order to find the best possible value of σ for each synthetic dataset, we generate 100 dataset for each value τ (the parameter that indicates the separation between cluster means) considered in our simulation setting, which are: $\tau = 1.5$ in setting A (similar datasets); $\tau = 0, 1, 2, 3$ in setting B (datasets with different levels of noise); $\tau = 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4$ in the additional simulation settings presented below (Section B.1.2). For each dataset, we build one kernel for each of the following values of σ: 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 15. We then use kernel k-means to cluster the data and compute the ARI between the clustering obtained in this way and the true cluster labels (Figure B.2). Finally, we choose the value of σ maximising the average ARI for each value of τ.

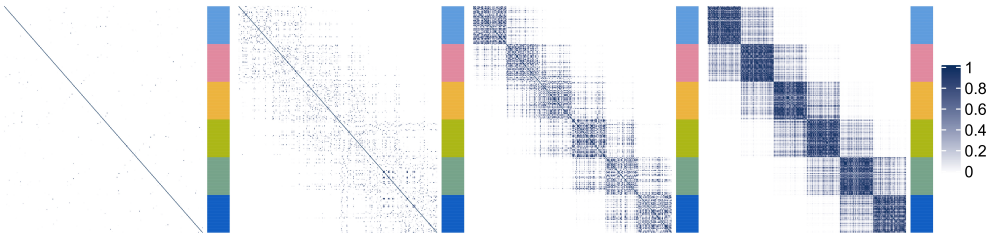


FIGURE B.1: Kernels obtained for the same datasets as those used for Figure 3.3 using RBF kernels.

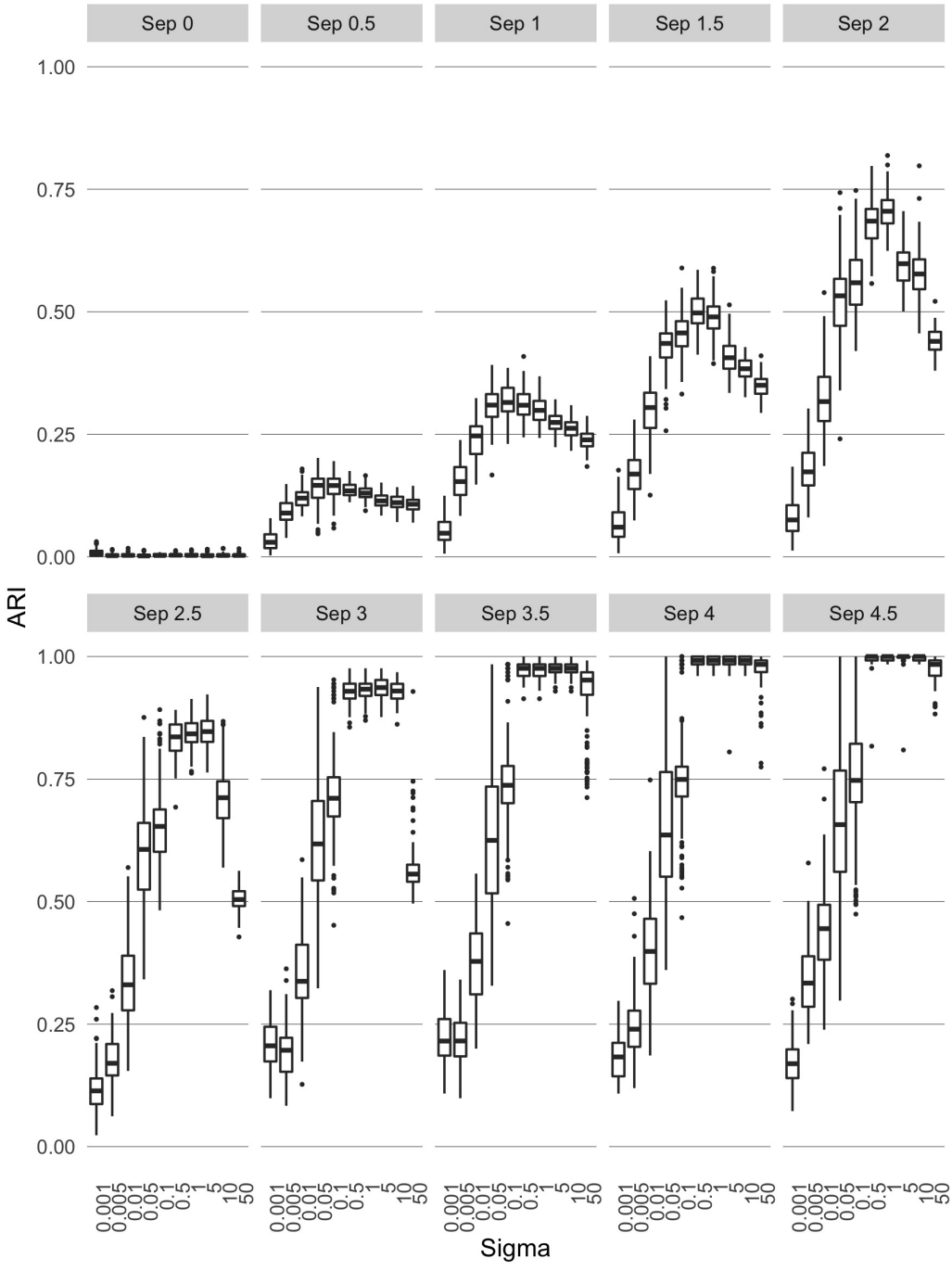
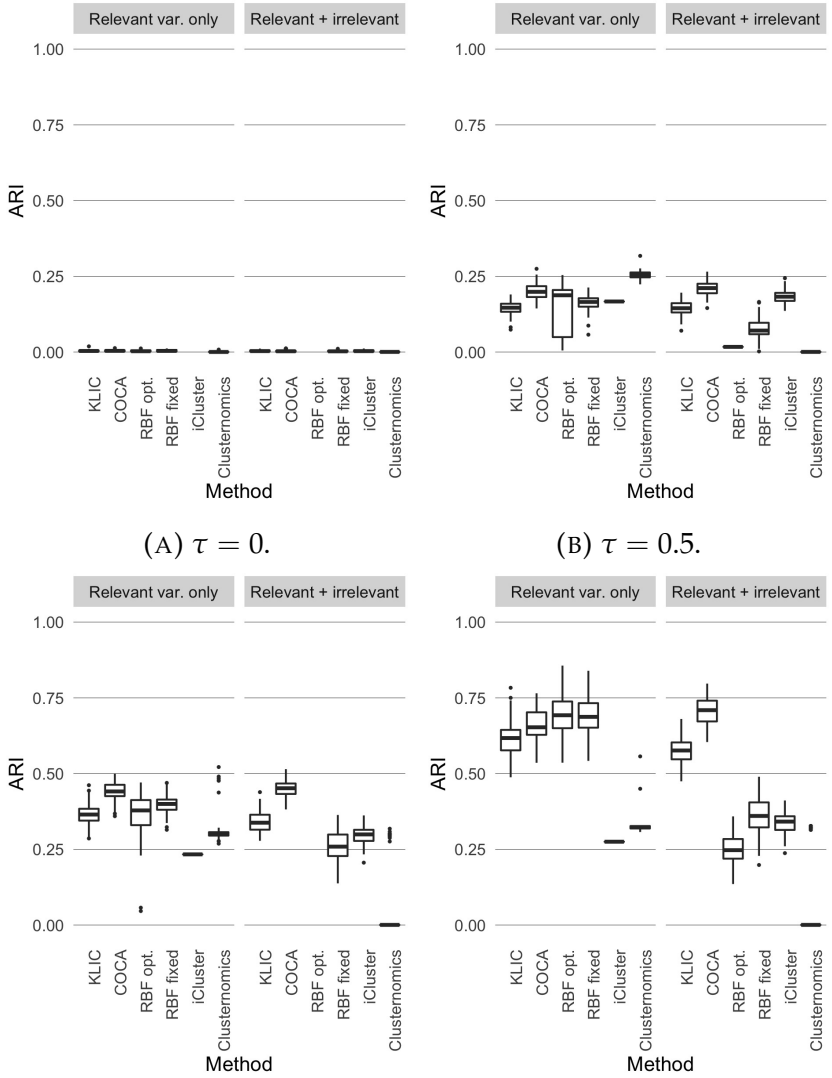


FIGURE B.2: ARI between the clusters obtained with kernel k-means on RBF kernels for different values of the characteristic length scale parameter and separation between clusters.

B.1.2 *Additional simulation settings*

For simulation setting A (four datasets with the same level of cluster separability) only the results obtained with $\tau = 1.5$ are reported in the main paper. For completeness, we show here the corresponding figures for a range of other values of τ in Figures B.3 and B.4.



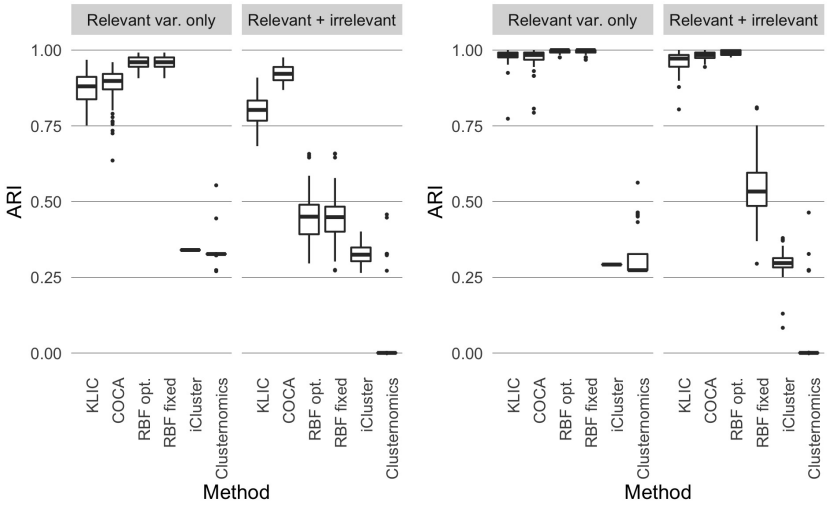
(A) $\tau = 0$.

(B) $\tau = 0.5$.

(C) $\tau = 1$.

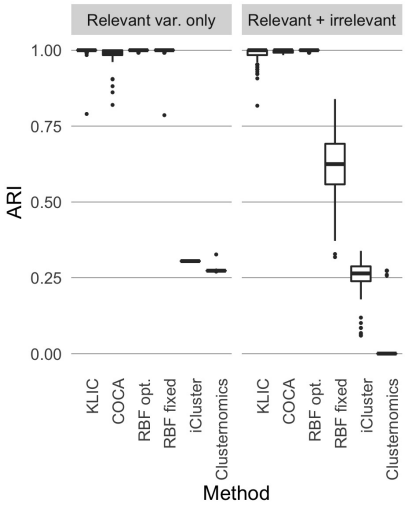
(D) $\tau = 1.5$.

FIGURE B.3: Comparison between KLIC, COCA, and other clustering algorithms. ARI obtained using four datasets having the same clustering structure and cluster separability (as in Figure 3.4).

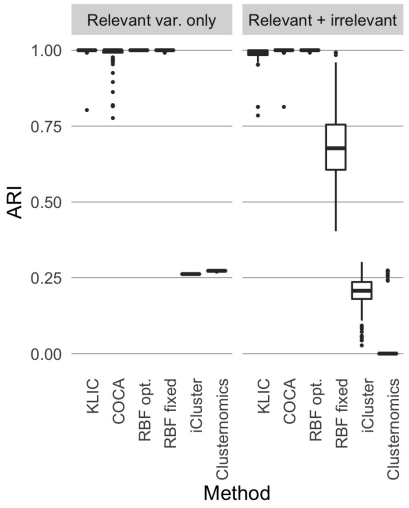


(A) $\tau = 2$.

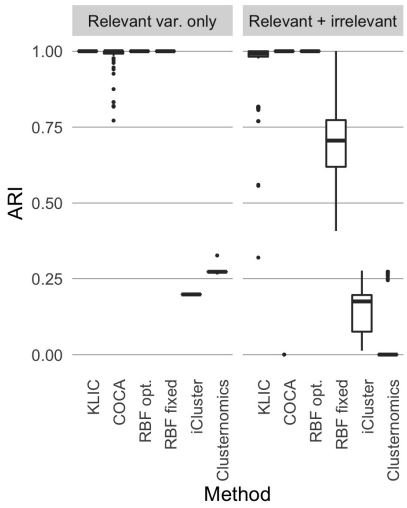
(B) $\tau = 2.5$.



(C) $\tau = 3$.



(D) $\tau = 3.5$.



(E) $\tau = 4$.

FIGURE B.4: Comparison between KLIC, COCA, and other clustering algorithms. ARI obtained using four datasets having the same clustering structure and cluster separability (as in Figure 3.4).

B.2 MULTIPLATFORM ANALYSIS OF 12 CANCER TYPES

In Section B.2.1 we explain the steps we took to try to replicate the data preprocessing and cluster analysis of Hoadley et al., 2014. In Section B.2.2 we give more details on the input and output of KLIC for this particular application.

B.2.1 *Replicating the analysis of Hoadley et al., 2014*

For each type of data we followed as closely as possible the procedures presented in the supplementary material of Hoadley et al., 2014. We present here the steps that we followed. The agreement between the clustering analysis presented here and the clustering presented in the original Hoadley *et al.* paper ranged from excellent (for the protein and mRNA datasets) to quite poor (for the miRNA dataset).

Protein expression We used hierarchical clustering with Ward's agglomeration method and Pearson's correlation as the distance. Our clusters match exactly those of Hoadley *et al.* (i.e. the ARI is equal to one, see Figure B.5).

mRNA expression For mRNA expression, we proceeded as indicated by Hoadley et al., 2014. We chose the genes present in 70% of samples and then selected the 6,000 most variable genes. Then we used the ConsensusClusterPlus R package with settings `maxK=20`, `innerLinkage ="average" finalLinkage="average"`, `distance="pearson"`, and `corUse= "pairwise.complete.obs"`. The ARI is 0.917 (see Figure B.6).

DNA methylation We used hierarchical clustering with Jaccard's distance and Ward's agglomeration method. Hoadley et al., 2014 chose to divide the data into 19 clusters, so we did the same. Comparing our clusters to those of Hoadley et al., 2014, we obtained an ARI of 0.680 (see Figure B.7).

DNA copy number The clusters for the somatic copy number dataset were found using hierarchical clustering with Euclidean distance and Ward's method. The number of clusters was set to eight in the original manuscript based on the cophenetic distances and therefore we did the same here. The adjusted Rand index (ARI) comparing the clustering found in the present analysis with the clustering found in the original analysis of Hoadley *et al.* is 0.333 (see Figure B.8).

miRNA expression In the original manuscript the clusters of the miRNA-seq data were determined using a software program called *Cluster 3* (De Hoon et al., 2004). The same software was used to scale the data. Since it is was not possible to retrieve the clusters presented in the paper using this software, we

used R to scale the data as was done by Cluster 3, namely applying a logarithmic transformation to the data and then median-centring. We found the final clusters using agglomerative hierarchical clustering in R (`agnes` command). We selected the number of clusters that maximises the silhouette, which is eight. The ARI is 0.255 (see Figure B.9).

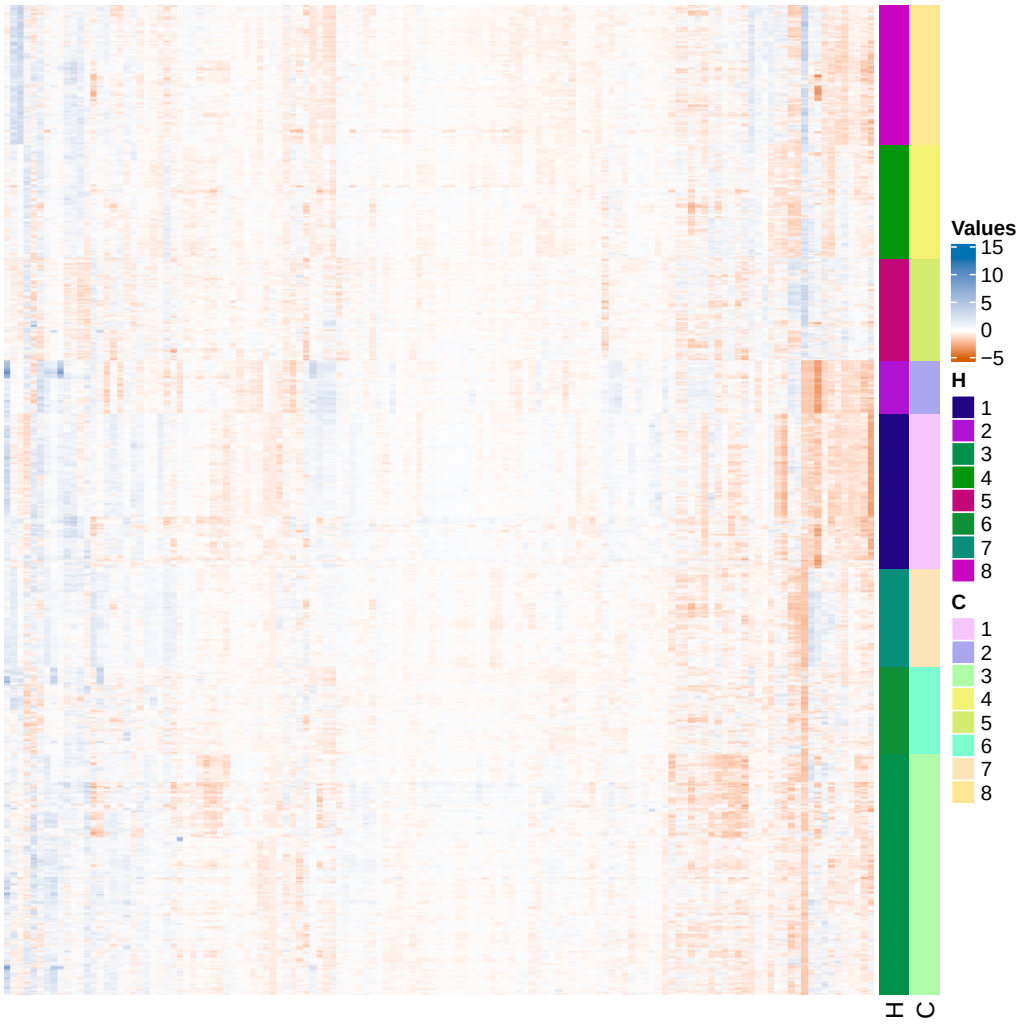


FIGURE B.5: Protein expression clusters. High values are indicated in blue and low values in orange. C: Clusters found in this analysis. H: Clusters found in the original analysis of Hoadley *et al.* ARI between C and H: 1.

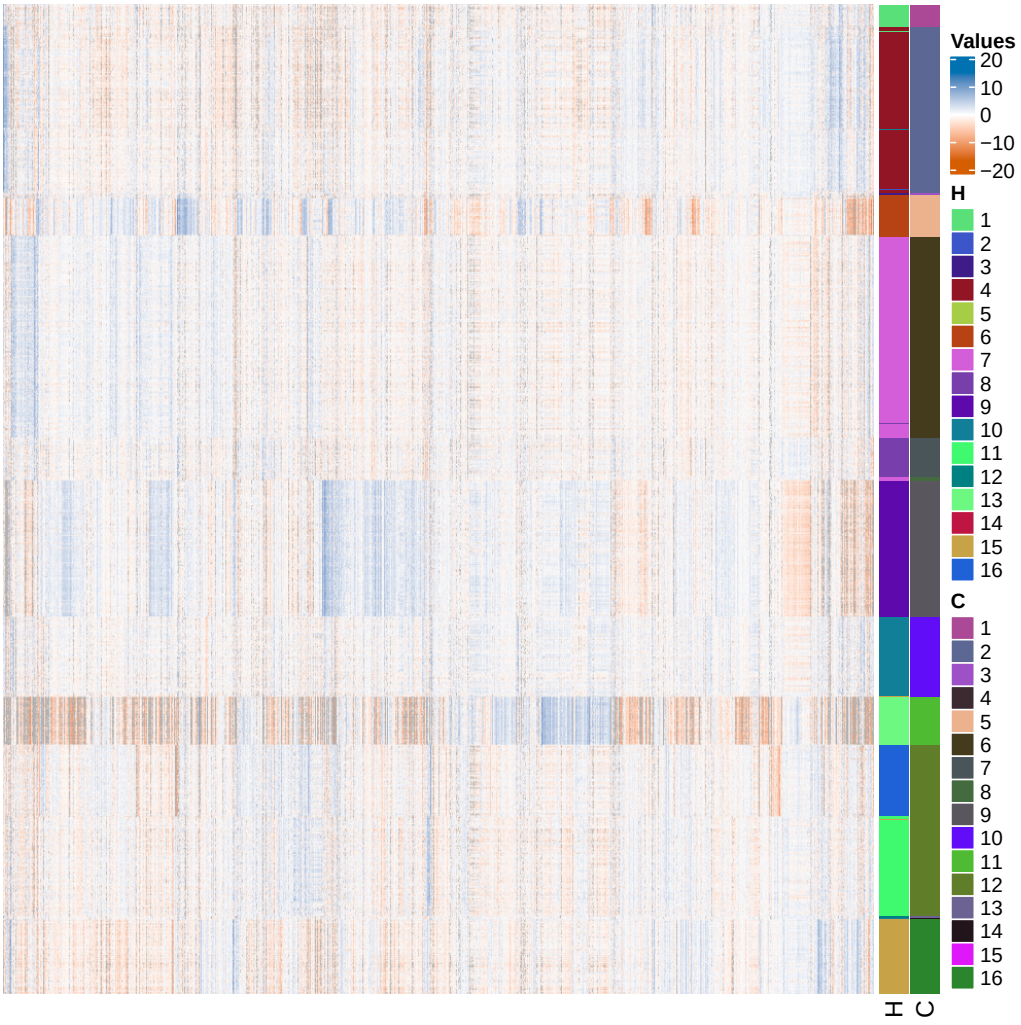


FIGURE B.6: mRNA expression clusters. High values are indicated in blue and low values in orange. The dataset contains 600 genes but here we show only 100 of them. C: Clusters found in this analysis. H: Clusters found in the original analysis of Hoadley *et al.* ARI between C and H: 0.917.

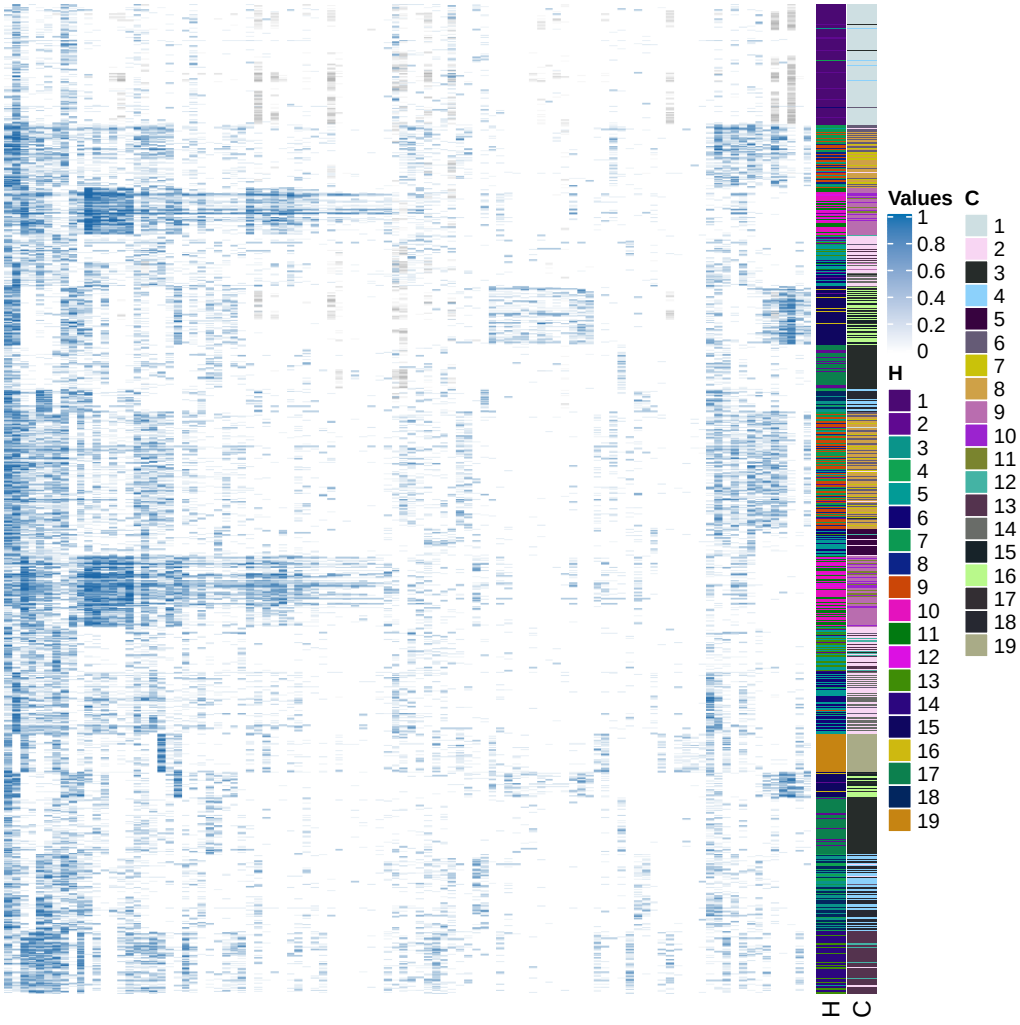


FIGURE B.7: DNA methylation clusters. Blue cells correspond to methylated loci. Missing values are indicated in grey colour. Only 100 CpG loci are shown here, but the full dataset contains 2,043. C: Clusters found in this analysis. H: Clusters found in the original analysis of Hoadley *et al.* ARI between C and H: 0.680.

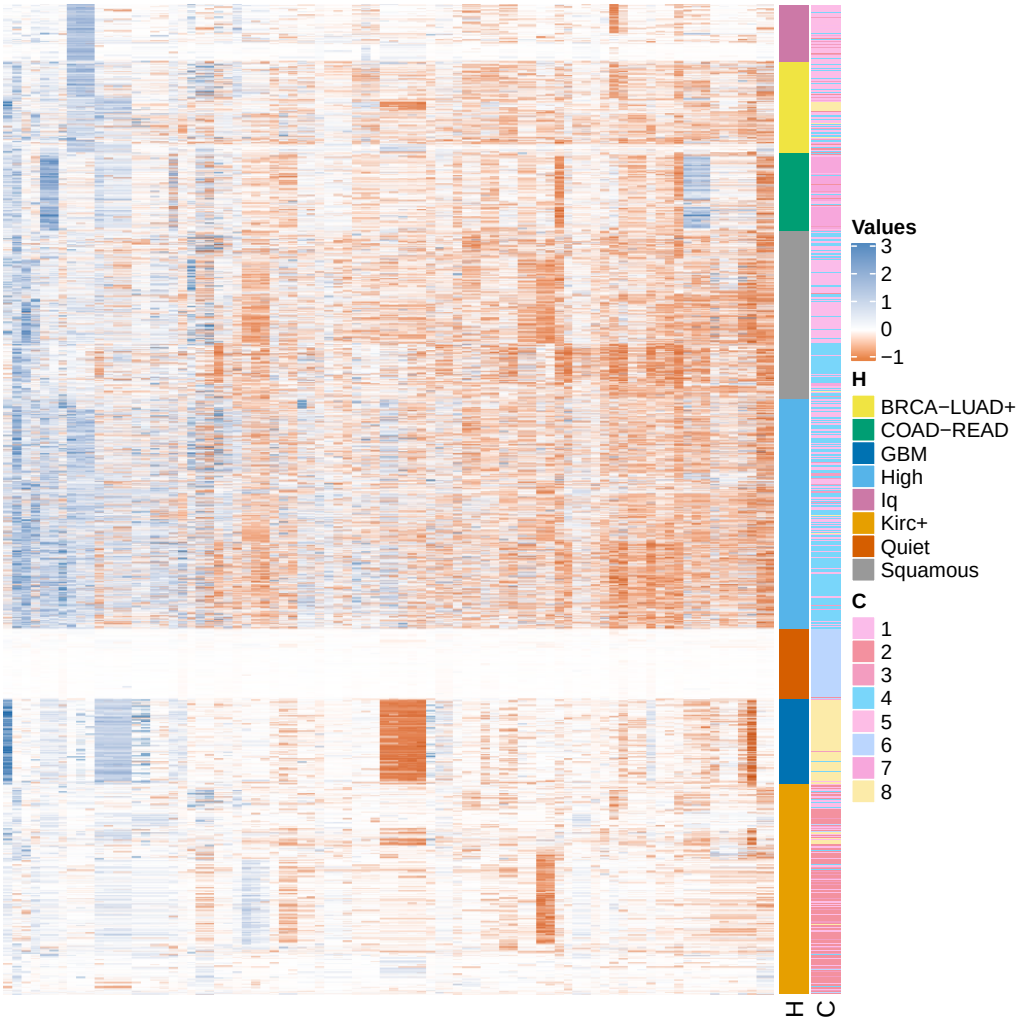
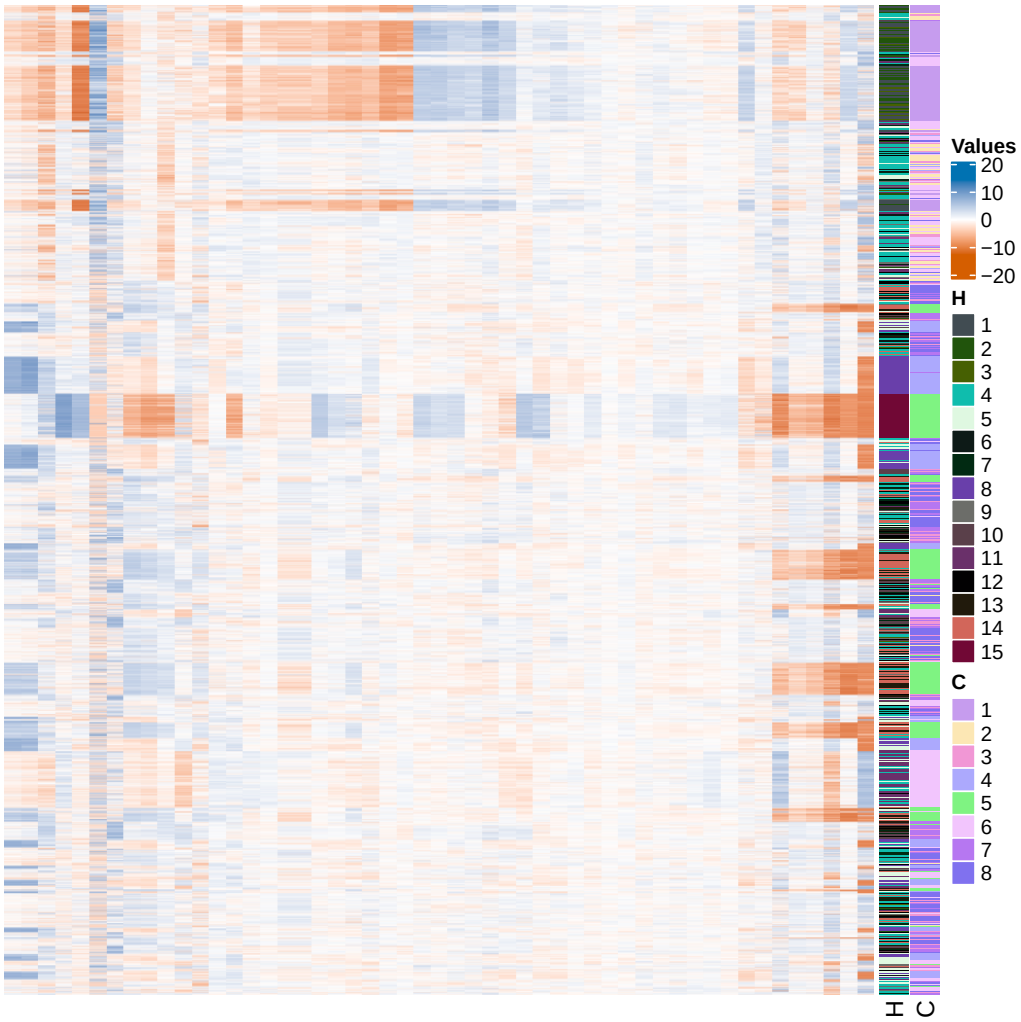
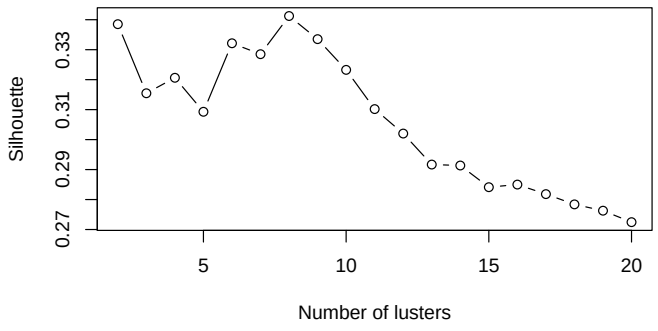


FIGURE B.8: Somatic copy number clusters. High values are indicated in blue and low values in orange. C: Clusters found in this analysis. H: Clusters found in the original analysis of Hoadley *et al.* ARI between C and H: 0.333.



(A) Clusters. High values are indicated in blue, low values in orange.

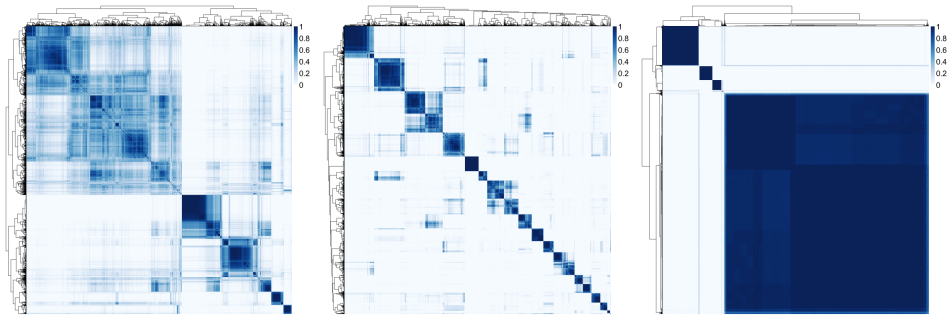


(B) Average silhouette.

FIGURE B.9: miRNA expression. C: Clusters found in this analysis. H: Clusters found in the original analysis of Hoadley *et al.* ARI between C and H: 0.255.

B.2.2 *Output of KLIC*

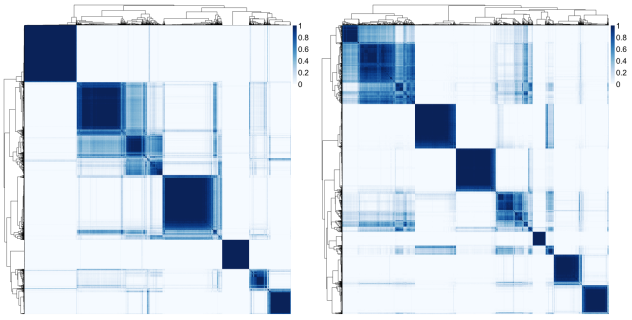
The kernels corresponding to each dataset are shown in Figure B.10, for each of them we also report the cophenetic correlation coefficient. Figure B.11a shows the weights associated to each observation in each dataset. Figure B.11b shows the correspondences between the clusters obtained using KLIC and the tumour tissues. Most clusters correspond quite well with one or two tissue types (e.g. cluster 10 contains almost exclusively samples of renal cell carcinoma and cluster 6 contains colon and rectal adenocarcinomas), but not all. Finally, Figure B.11c shows the average silhouette for all the number of clusters considered: the optimal values are between six and ten.



(A) DNA copy number. Cophenetic correlation coefficient: 0.736.

(B) DNA methylation. Cophenetic correlation coefficient: 0.859.

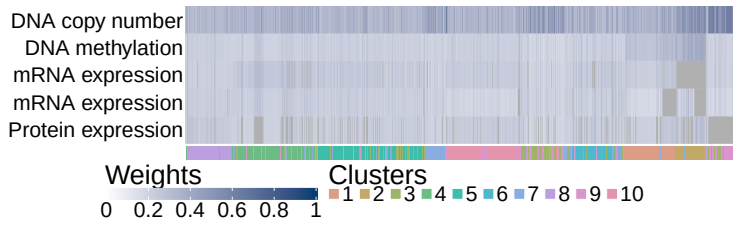
(C) mRNA expression. Cophenetic correlation coefficient: 0.974.



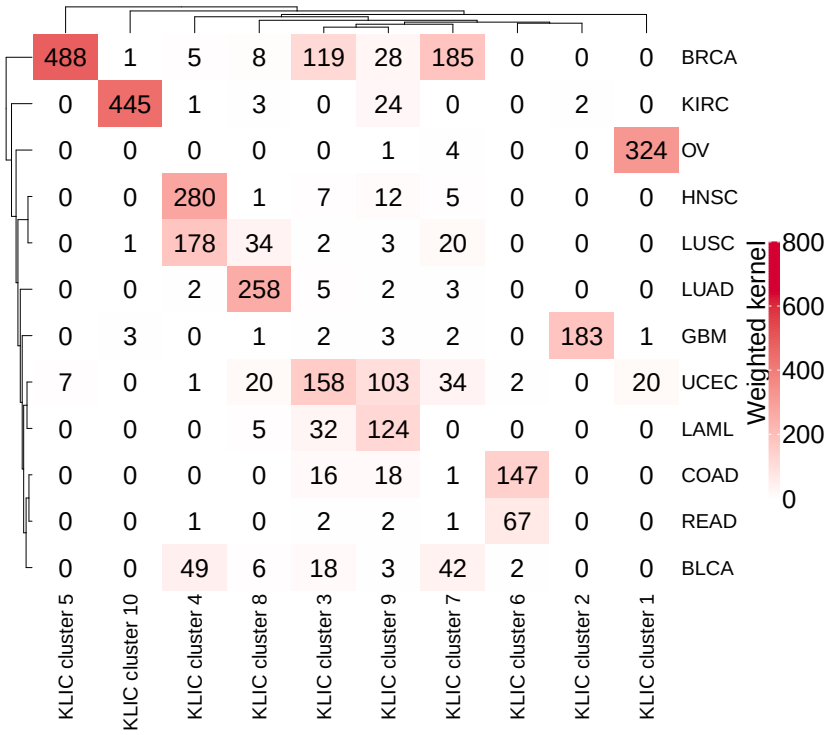
(D) miRNA expression. Cophenetic correlation coefficient: 0.923.

(E) Protein expression. Cophenetic correlation coefficient: 0.888.

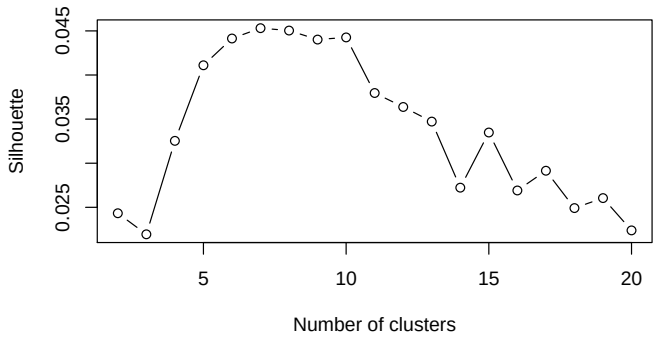
FIGURE B.10: Pan-cancer data: kernel matrices.



(A) Weights.



(B) Matrix of coincidences.



(C) Average silhouette.

FIGURE B.11: Multiplatform analysis of 12 cancer types, output of KLIC. (A) Weights. Low weights are indicated in white and higher weights in green. Grey cells correspond to missing values, which have zero weight. (B) Matrix showing the correspondences between the clusters obtained by using KLIC and the tumour tissues. (C) Average silhouette. The maximum is obtained for seven clusters. All numbers of clusters comprised between six and ten have similar values.

B.3 TRANSCRIPTIONAL MODULE DISCOVERY

This section is structured as follows. First, we give further details regarding the application of KLIC and COCA to transcriptional module discovery using Bayesian Hierarchical Clustering as the clustering algorithm for the ChIP data. Then, we consider other algorithms that could have been applied to this dataset and compare the new results with those reported in the main paper. Finally, we give more details about the choice of the number of clusters for PAM.

B.3.1 *Clustering algorithms for the ChIP data*

The ChIP dataset is quite sparse. The data were discretised so that only transcription factors that are believed with high confidence to be able to bind to a gene's promoter region are marked as "ones"; all the others are "zeros". For this reason, in addition to BHC, we considered two clustering algorithms that are able to take into account this feature of the data. However, we show that these methods often cluster genes with few transcription factors (i.e. observations for which most variables are zero) together, while the other genes end up in separate small clusters that are less stable under subsampling of the data. This leads to consensus matrices that have high cophenetic correlation coefficients but carry little information. We show that combining the corresponding kernels to that of the expression data does not always give more meaningful clustering solutions than those obtained on each data type separately. This highlights the importance of the kernel matrices as an intermediate diagnostic tool for KLIC, which can help choosing the right clustering algorithms.

Bayesian Hierarchical Clustering

Bayesian hierarchical clustering (Heller and Ghahramani, 2005) is a method for agglomerative hierarchical clustering. The idea is that, similarly to classical agglomerative clustering algorithms, at the start each data point is considered as a different cluster; then, at each step, two clusters are merged. The main difference between classical hierarchical clustering and BHC is that in BHC merging is done based on Bayesian hypothesis testing, where the alternative hypotheses are "all data in clusters c_i and c_j were generated from the same probabilistic model" and "the data in c_i and c_j has two or more clusters in it". The pair of clusters that is selected for merging is the one with highest probability of the merged hypothesis.

Figure B.12b shows the clusters found on all the data (on the left) as well as the consensus matrix obtained by applying BHC to 200 random subsamples of 95% of the data. This shows that, while the clustering algorithm works well on the full dataset, different clustering structures are found in the data subsamples, giving a fuzzy similarity matrix. This is due to the fact that most clusters are very small,

and are hard to identify when only a subset of the data is available. The output of COCA obtained with this clustering algorithm is shown in Figure B.13, the output KLIC is shown in the main paper. Higher weights are assigned on average to the expression data, with an average of 0.58.

PAM with Gower's distance

Another clustering algorithm that could have been applied to this dataset is PAM with Gower's distance (Gower, 1971). In this case, all variables are binary and therefore Gower's distance is equivalent to Jaccard's distance. For two multivariate binary observations x_i and x_j, this is defined as one minus the Jaccard index:

$$J = \frac{M_{11}}{M_{01} + M_{01} + M_{11}},$$

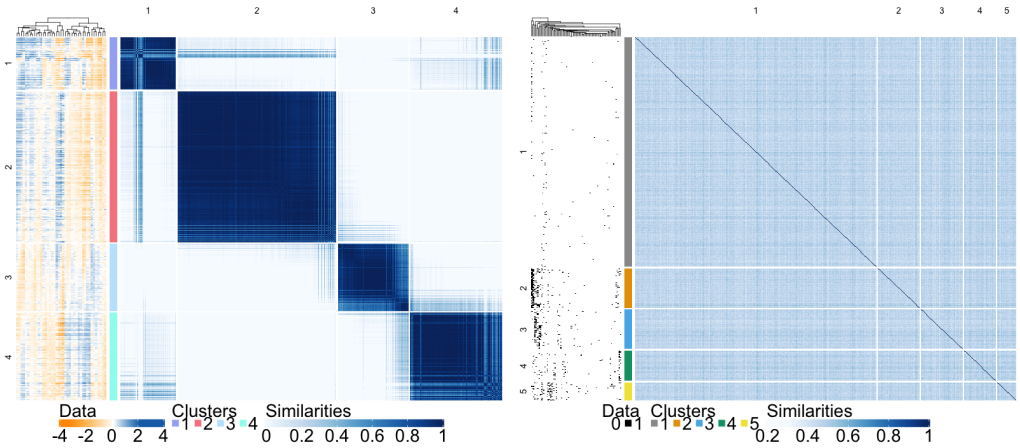
where M_{11} is the number of variables where x_i and x_j both have value of 1, M_{01} is the number of variables where x_i is 0 and x_j is 1 and viceversa for M_{01}. This distance is particularly suited for this dataset because here the ones correspond to transcription factors that are believed with high confidence to be able to bind to the promoter region of the corresponding gene, whereas zeros are transcription factors for which we are not able to reject the hypothesis that they do not bind to that promoter region. Thus, in a sense, ones carry more information than zeros.

The consensus matrix obtained by subsampling 200 times 95% of the data is shown in Figure B.12c, the output of COCA and KLIC in Figures B.13 and B.14 respectively. Details on how the number of clusters was chosen are given in Section B.3.2. As usual, the number of clusters for KLIC and COCA was chosen in order to maximise the silhouette. KLIC selected $K = 3$ and COCA $K = 10$. GOTO scores for the clustering found with PAM algorithm and Gower's distance, as well as those given by KLIC and COCA for three and ten clusters are reported in Table B.1. Higher weights are assigned to the ChIP data, with an average of 0.78.

Greedy Bayesian non-parametric clustering algorithm

The last clustering algorithm that we considered is a greedy approximation to the Gibbs sampling algorithm for Dirichlet process mixture models of Neal, 2000. In the greedy version of the algorithm used here at each iteration cluster allocations are made in a deterministic fashion, assigning each observation to the cluster with highest probability, instead of sampling the cluster labels according to their conditional probabilities.

Figure B.12d shows the consensus matrix, Figures B.13 and B.14 show the output of COCA and KLIC respectively. (Note that, for brevity, we refer to this method as GBNP, which stands for greedy Bayesian non-parametric algorithm.) Higher weights are assigned to the ChIP data points, with an average of 0.59.



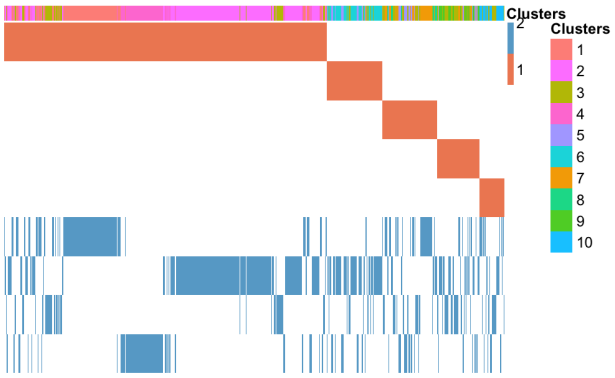
(A) Expression data, PAM. Cophenetic correlation coefficient: 0.971.

(B) ChIP data, BHC. Cophenetic correlation coefficient: 0.103.

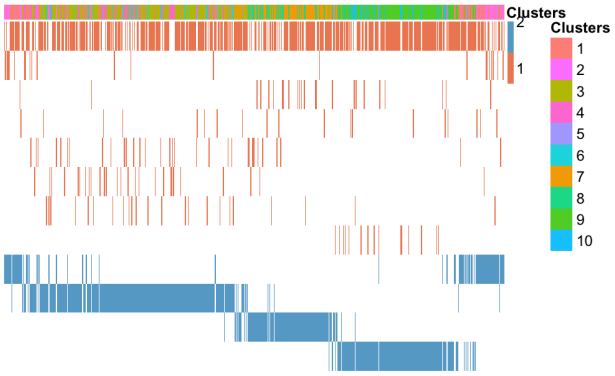
(C) ChIP data, PAM. Cophenetic correlation coefficient: 0.996.

(D) ChIP data, GBNP. Cophenetic correlation coefficient: 0.931.

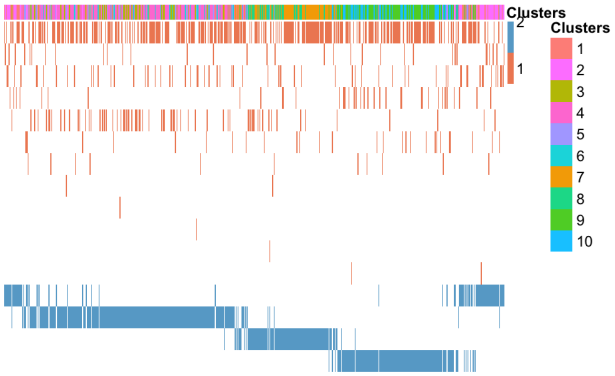
FIGURE B.12: Transcriptional module discovery. Left: original data, where each row is a gene and each column a feature. The rows are sorted by cluster. Centre: cluster obtained on the data. Right: consensus matrices, where each row and column corresponds to a gene.



(A) BHC

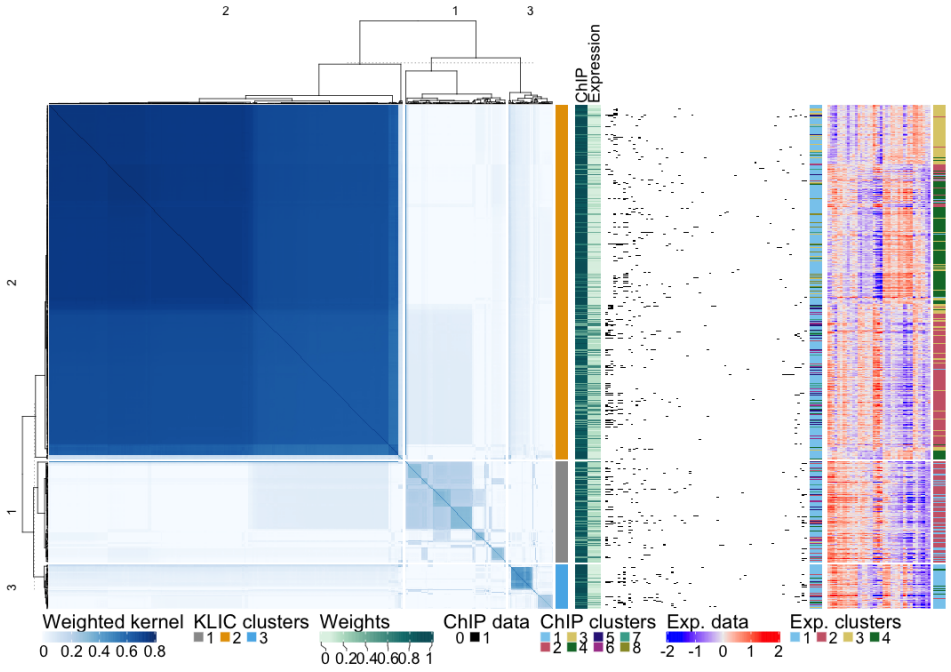


(B) PAM with Gower's distance.

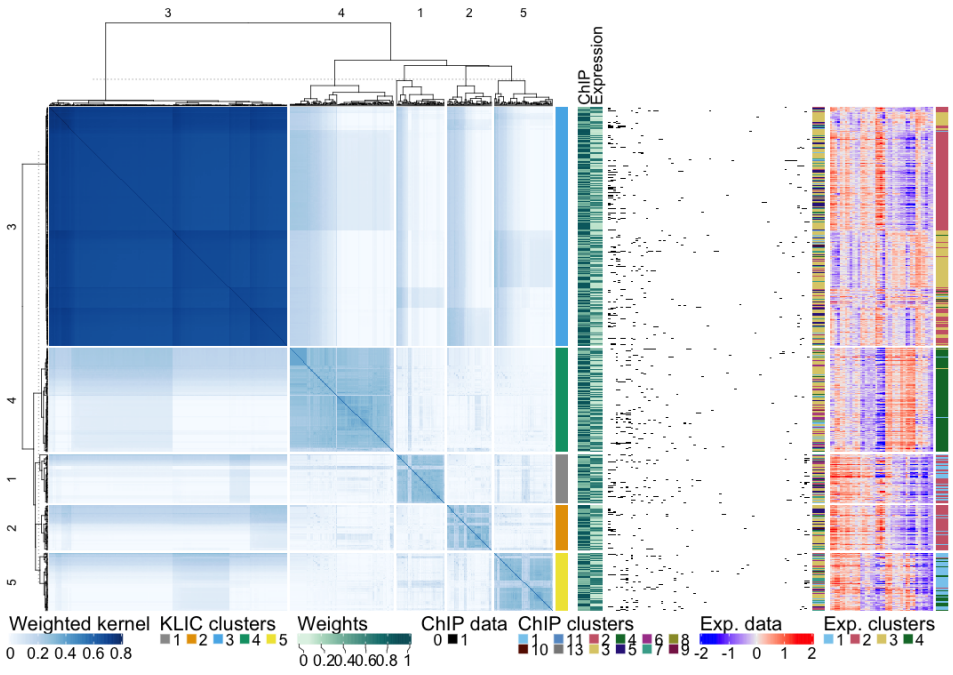


(C) GBNP.

FIGURE B.13: Transcriptional module discovery, output of COCA. MOCs and final clusters obtained using different clustering algorithms to generate the consensus matrix of the transcription factor data.



(A) PAM with Gower's distance



(B) GBNP

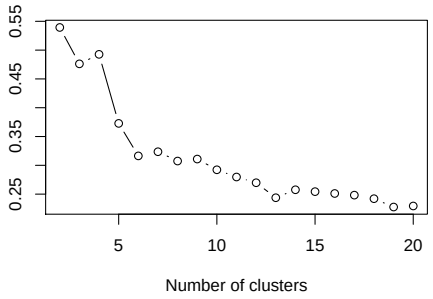
FIGURE B.14: Transcriptional module discovery, output of KLIC. Weighted similarity matrices obtained using different clustering algorithms to generate the consensus matrix of the transcription factor data. To the right of each similarity matrix are shown the final clusters, the weights assigned to each observation in each cluster by multiple kernel k-means, the original datasets, and the clusters obtained on each dataset.

Clusters	Dataset(s)	Algorithm	GOTO BP	GOTO MF	GOTO CE
4	Expression	PAM correlation	6.1194	0.9075	8.4139
8	ChIP	PAM Gower's	6.0872	0.8959	8.3261
5	ChIP	BHC	6.0020	0.9192	8.2886
12	ChIP	GBNP	6.0192	0.9176	8.3664
4	ChIP + Expression	COCA (PAM + BHC)	6.1194	0.9075	8.4139
4	ChIP + Expression	KLIC (PAM + BHC)	6.1221	0.9074	8.4103
10	ChIP + Expression	COCA (PAM + BHC)	6.2767	0.9347	8.5137
10	ChIP + Expression	KLIC (PAM + BHC)	6.3240	0.9473	8.5310
3	ChIP + Expression	COCA (PAM + PAM)	5.9609	0.8991	8.2780
3	ChIP + Expression	KLIC (PAM + PAM)	5.9188	0.8915	8.1766
10	ChIP + Expression	COCA (PAM + PAM)	6.3429	0.9211	8.5126
10	ChIP + Expression	KLIC (PAM + PAM)	6.3724	0.9094	8.4868
5	ChIP + Expression	COCA (PAM + GBNP)	6.1298	0.9078	8.4218
5	ChIP + Expression	KLIC (PAM + GBNP)	5.9629	0.9108	8.3246
10	ChIP + Expression	COCA (PAM + GBNP)	6.1605	0.9118	8.4796
10	ChIP + Expression	KLIC (PAM + GBNP)	6.2277	0.9262	8.4814

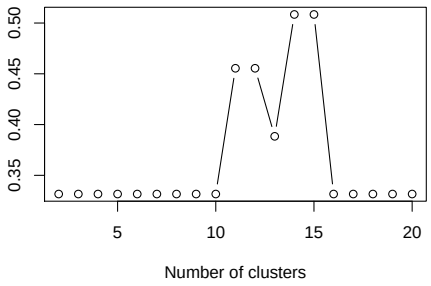
TABLE B.1: Gene ontology term overlap scores for different sets of data, clustering algorithms and numbers of clusters. BP stands for biological process ontology, MF for molecular function, and CE for cell component.

B.3.2 *Choice of the number of clusters*

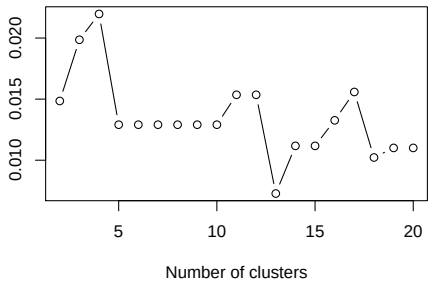
In order to choose the number of clusters when using PAM, we considered multiple metrics: the average silhouette (Rousseeuw, 1987), the gap statistic (Tibshirani, 2001), and the original and modified versions of Dunn's index (Dunn, 1974; Halkidi, Batistakis, and Vazirgiannis, 2001). We considered all number of clusters from two to 20. These are shown in Figures B.15 and B.16. For the expression data, we chose four clusters since three of the chosen metrics have a peak at $K = 4$. For the ChIP data, there is no consensus among the metrics, so we selected $K = 8$ based on the gap metric.



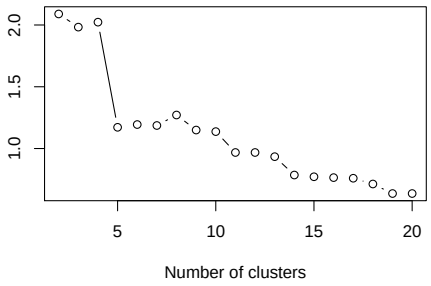
(A) Average silhouette.



(B) Widest gap.

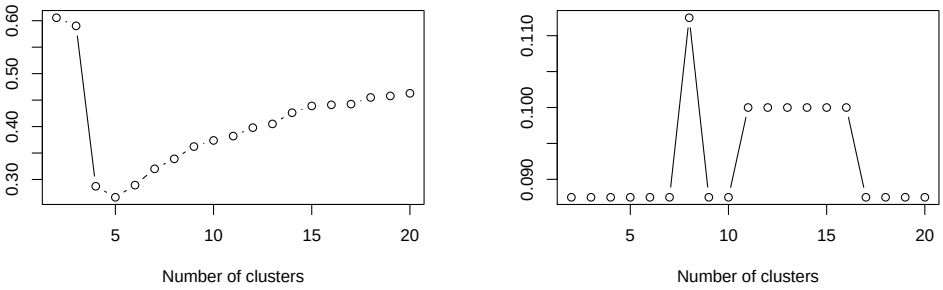


(C) Dunn's index.



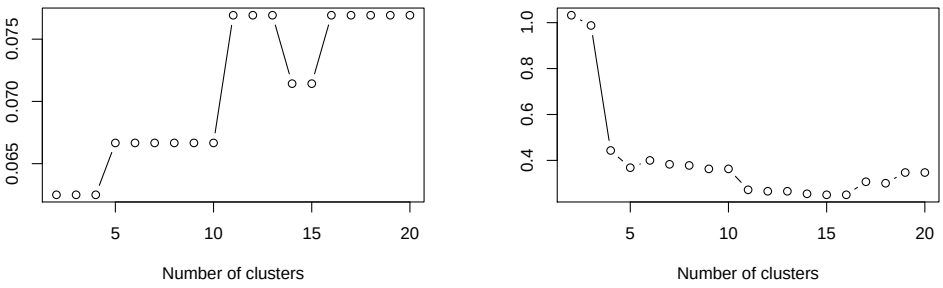
(D) Dunn's modified index.

FIGURE B.15: Expression data. Metrics used to choose the number of clusters.



(A) Average silhouette.

(B) Widest gap.



(C) Dunn's index.

(D) Dunn's modified index.

FIGURE B.16: ChIP data. Metrics used to choose the number of clusters.

C

APPENDIX TO CHAPTER 4

This appendix contains additional figures and results for the simulation studies and biological applications presented in Chapter 4.

C.1 SIMULATION STUDY

In Sections C.1.2 and C.1.2 are presented additional figures related to the simulation studies of Section 4.3. In Section C.1.3 are explored new simulation settings.

C.1.1 *Synthetic data*

In Figure C.1 is shown one of the datasets used for the simulation studies of Section 4.3, with value of w set to 0.8 and number of covariates equal to 20. The first ten covariates determine six different clusters, the remaining covariates have no clustering structure. The response variable is generated as described in Section 4.3.

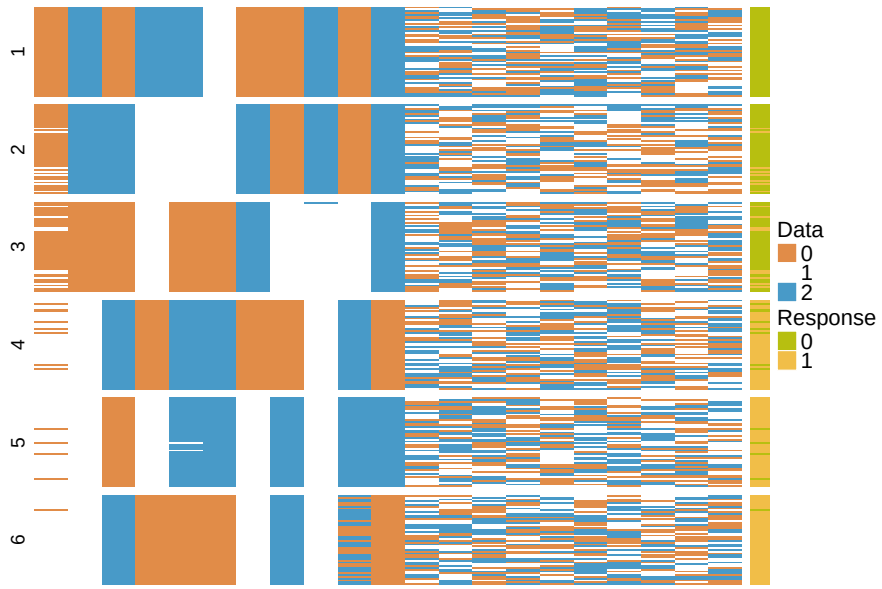


FIGURE C.1: One of the datasets used for the simulation studies. The data are categorical, taking values 0, 1, or 2, the response is binary. The rows are separated by cluster, the cluster labels are indicated on the right of the data matrix.

C.1.2 *Integrative clustering*

Figures C.2 and C.3 show the weights assigned to each dataset by the unsupervised and outcome-guided methods for the integration of multiple PSMs presented in the main paper.

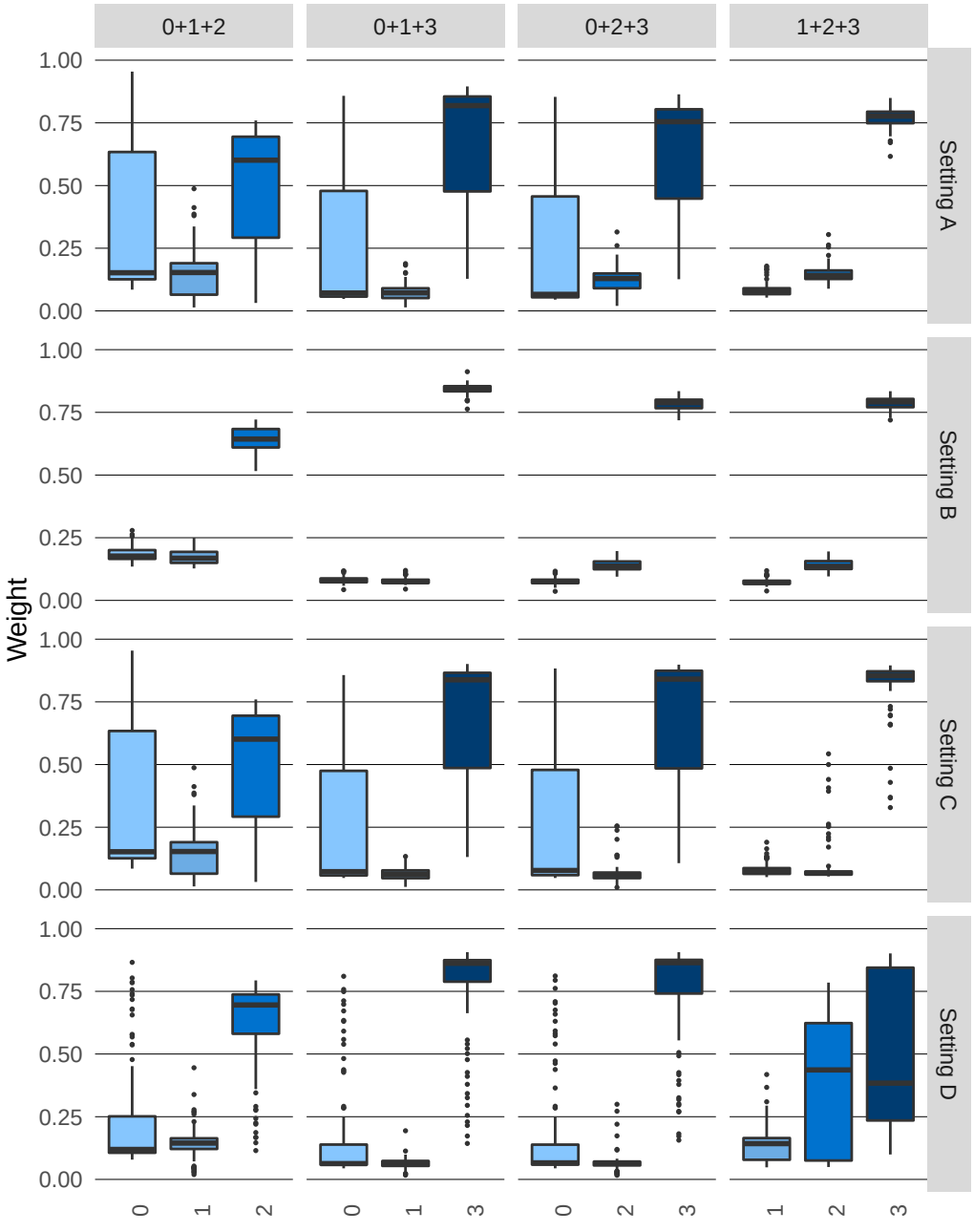


FIGURE C.2: Weights assigned to each PSM for each subset of datasets by the unsupervised integration method.

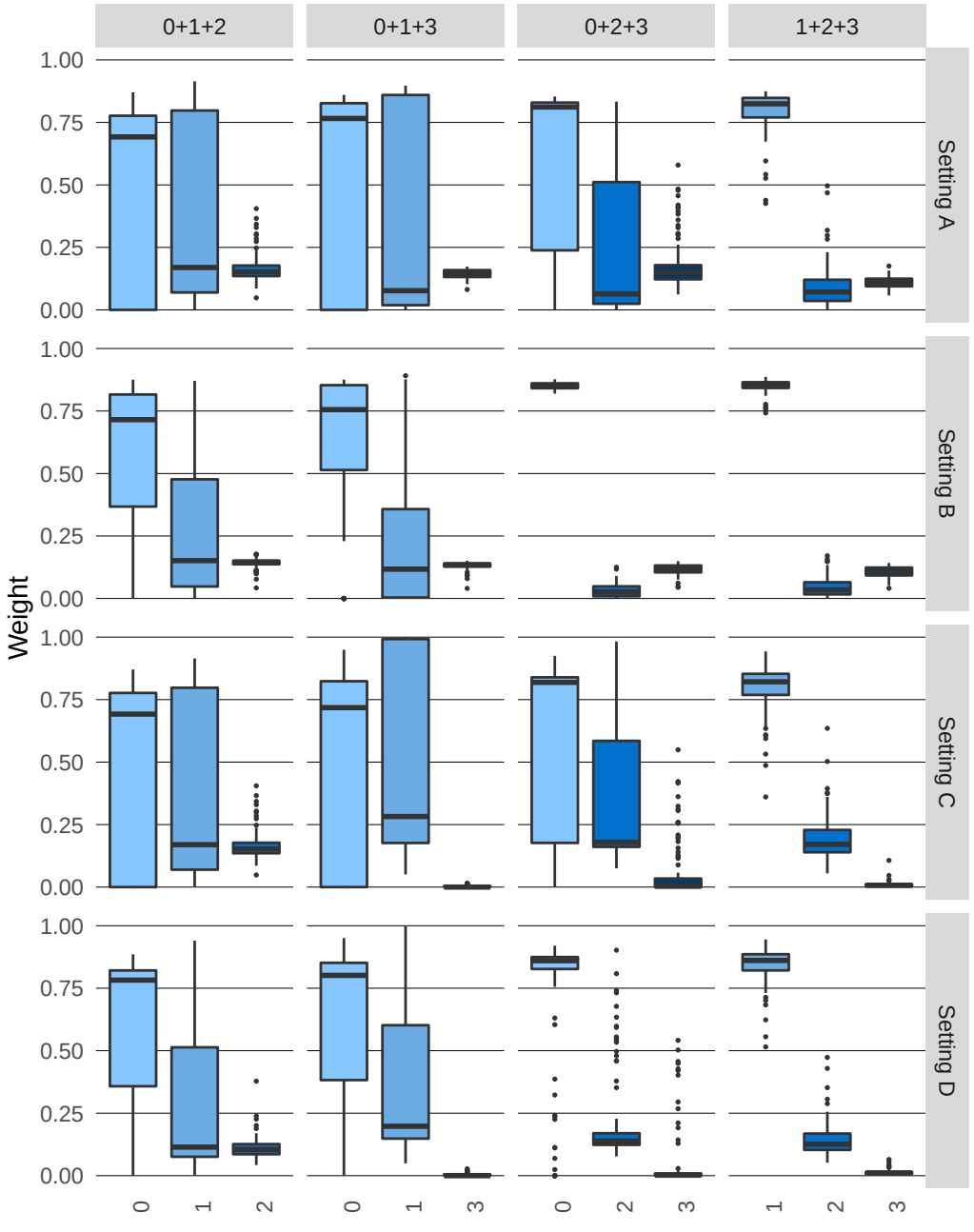


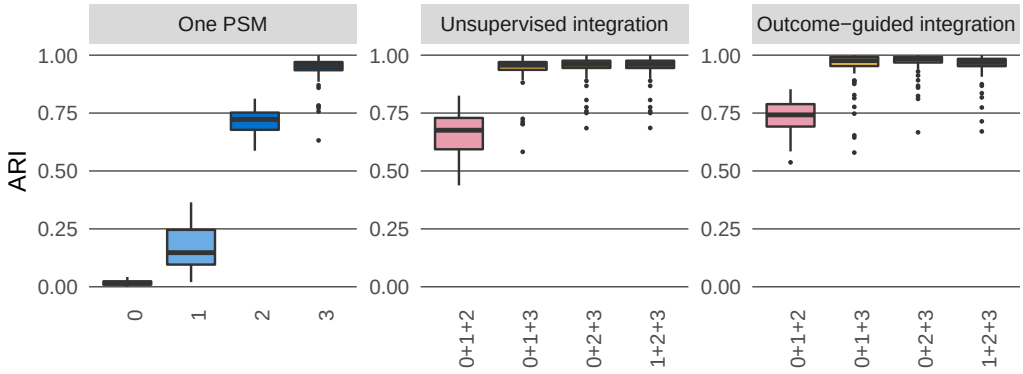
FIGURE C.3: Weights assigned to each PSM for each subset of datasets by the outcome-guided integration method.

C.1.3 *Additional simulation settings*

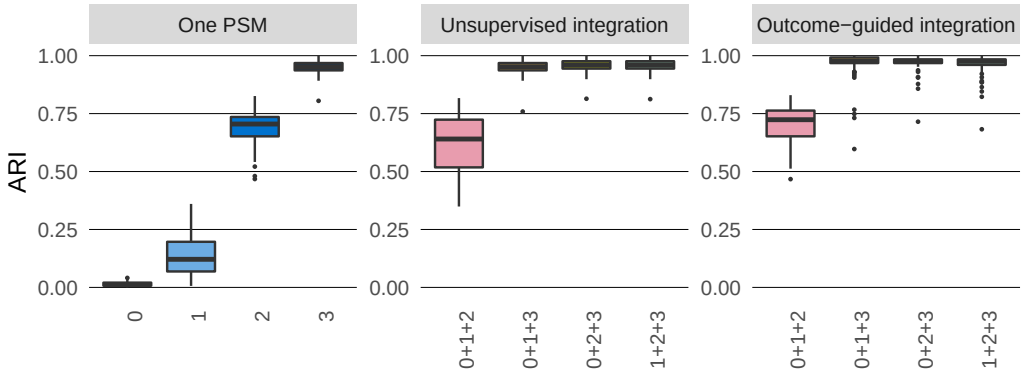
First, we consider a variation of simulation setting B of Section 4.3. Then we repeat the simulation studies of Chapter 3 with a different response variable.

Different number of covariates

We present here the results obtained for simulation setting B, with different numbers of irrelevant covariates. Figure C.4 shows the adjusted Rand index and Figure C.5 the weights assigned to each kernel.

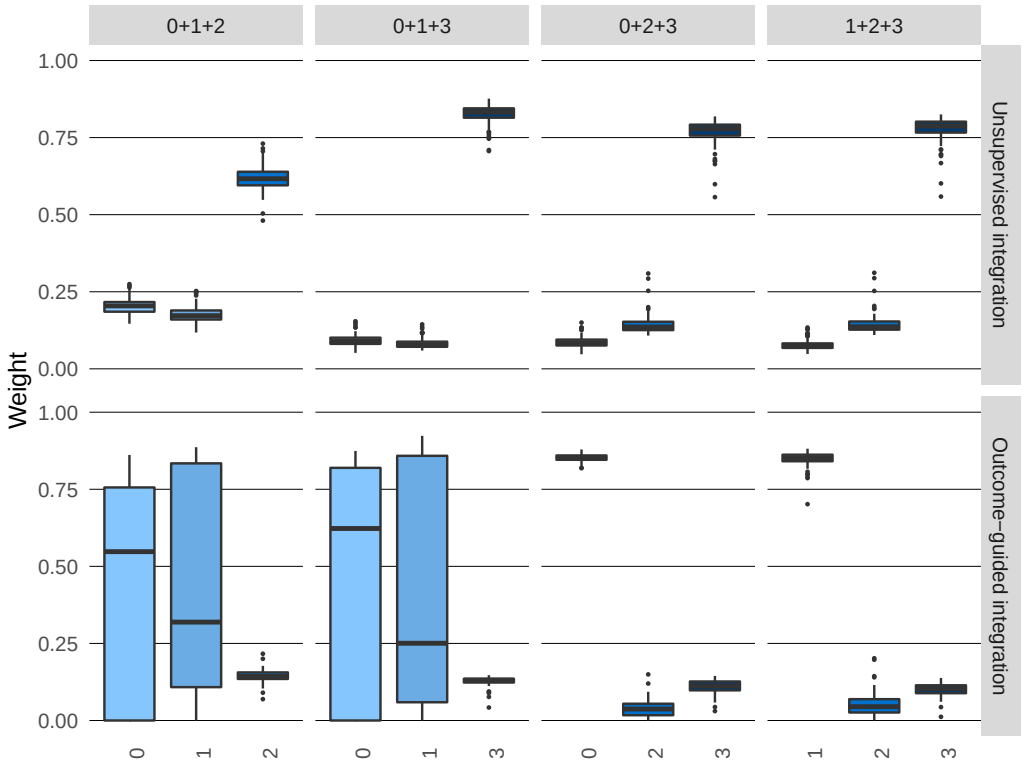


(A) Setting B, 2 covariates without clustering structure.

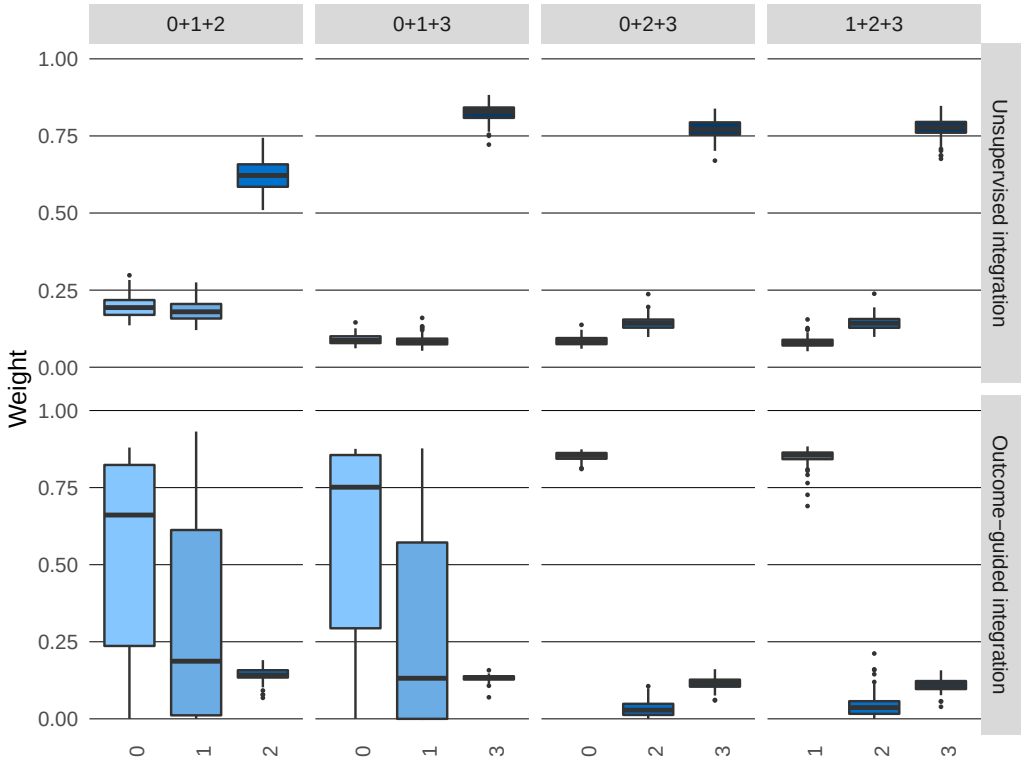


(B) Setting B, 5 covariates without clustering structure.

FIGURE C.4: ARI obtained by summarising the PSMs one at a time using kernel k-means (left), combining different subsets of three PSMs in an unsupervised fashion using localised multiple kernel k-means (centre), and combining the same subsets making use of a response variable and multiclass SVMs to determine each PSM's weight and using kernel k-means for the final clustering (right).



(A) Setting B, 2 covariates without clustering structure.



(B) Setting B, 5 covariates without clustering structure.

FIGURE C.5: Weights assigned to each PSM for each subset of datasets by the unsupervised (above) and outcome-guided (below) integration methods for PSMs.

Using the true cluster labels as response variable

We repeat the simulation study presented in Chapter 4 using the true cluster labels as the response variable, both for the outcome-guided integration (in all simulation settings) and to generate the PSMs with profile regression (in Setting D). Although the true cluster labels are not available in practice, this simulation study is used here to determine a putative upper bound on the performances of outcome-guided integration.

The ARI is reported in Figure C.6. As expected, the outcome-guided integration has higher values of the ARI in all settings, compared to the case where the outcome is a binary variable. Moreover, in Setting D the ARI of each PSM taken individually is also higher here than in the other simulation study.

First, for completeness, we show the weights assigned to each kernel in the unsupervised setting (Figure C.7). However, only the weights of Setting D differ from the previous simulation study (Figure C.2), because in Settings A, B, and C the response variable is not used. The weights assigned to each kernel matrix in the outcome-guided case are shown in Figure C.8. These are of easier interpretation compared to those presented above (Figure C.3): on average, kernels originated from datasets with higher cluster separability have higher weights.

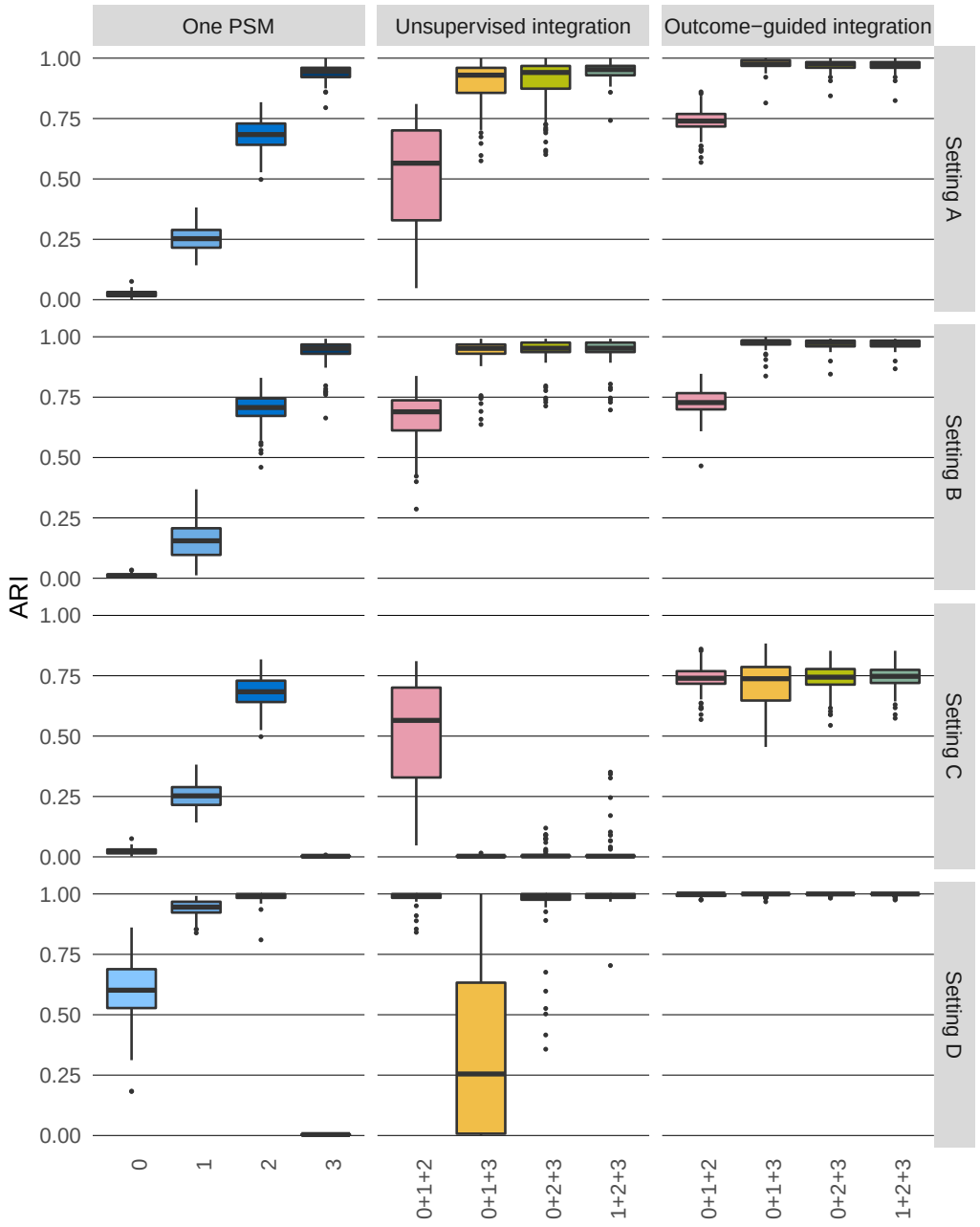


FIGURE C.6: Simulation study where the response for each observation is given by its true cluster label. ARI obtained by summarising the PSMs one at a time using kernel k-means (left), combining different subsets of three PSMs in an unsupervised fashion using localised multiple kernel k-means (centre), and combining the same subsets making use of a response variable and multi-class SVMs to determine each PSM's weight and using kernel k-means for the final clustering (right).

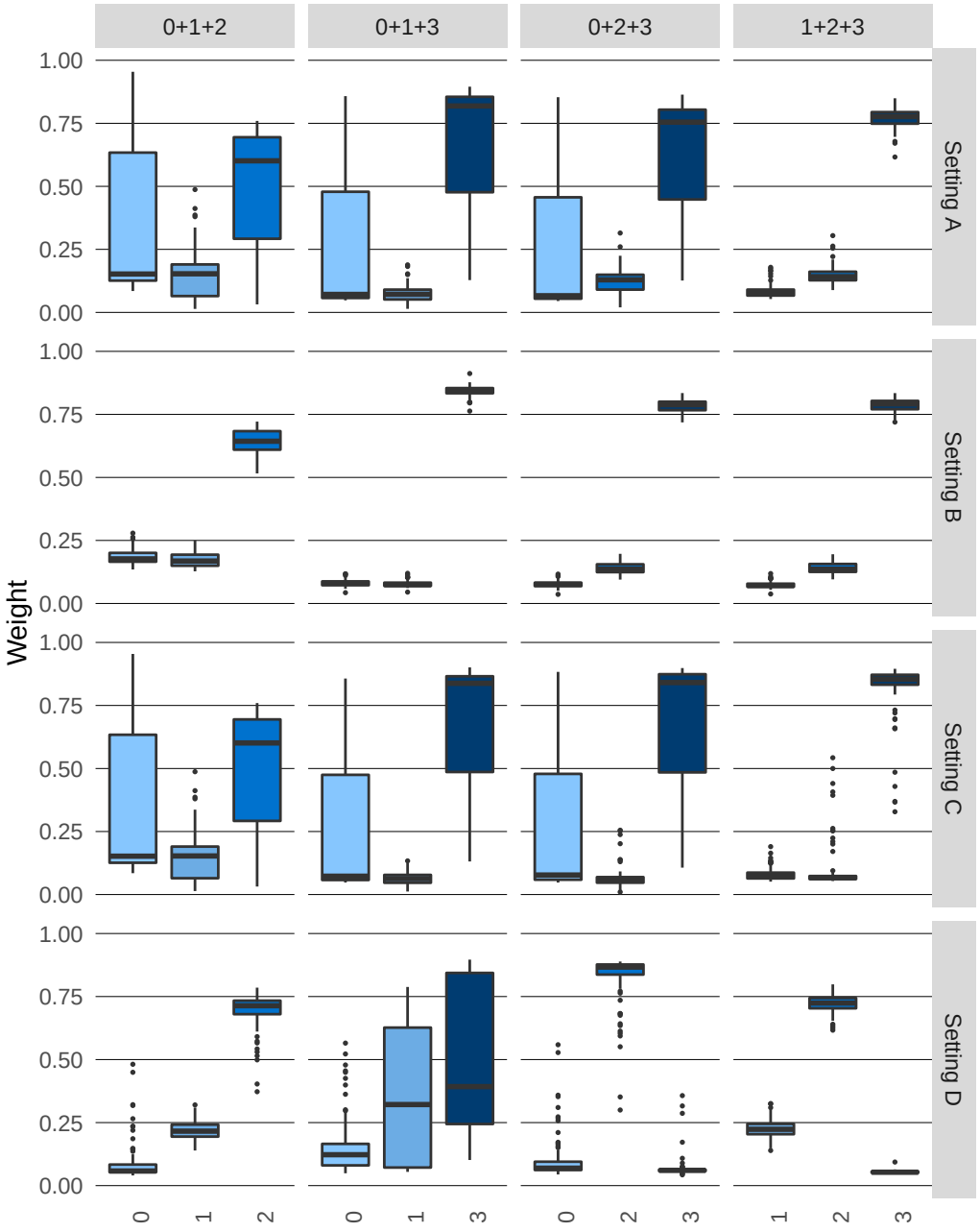


FIGURE C.7: Weights assigned to each PSM for each subset of datasets by the unsupervised integration method for the simulation study where the response for each observation is given by its true cluster label.

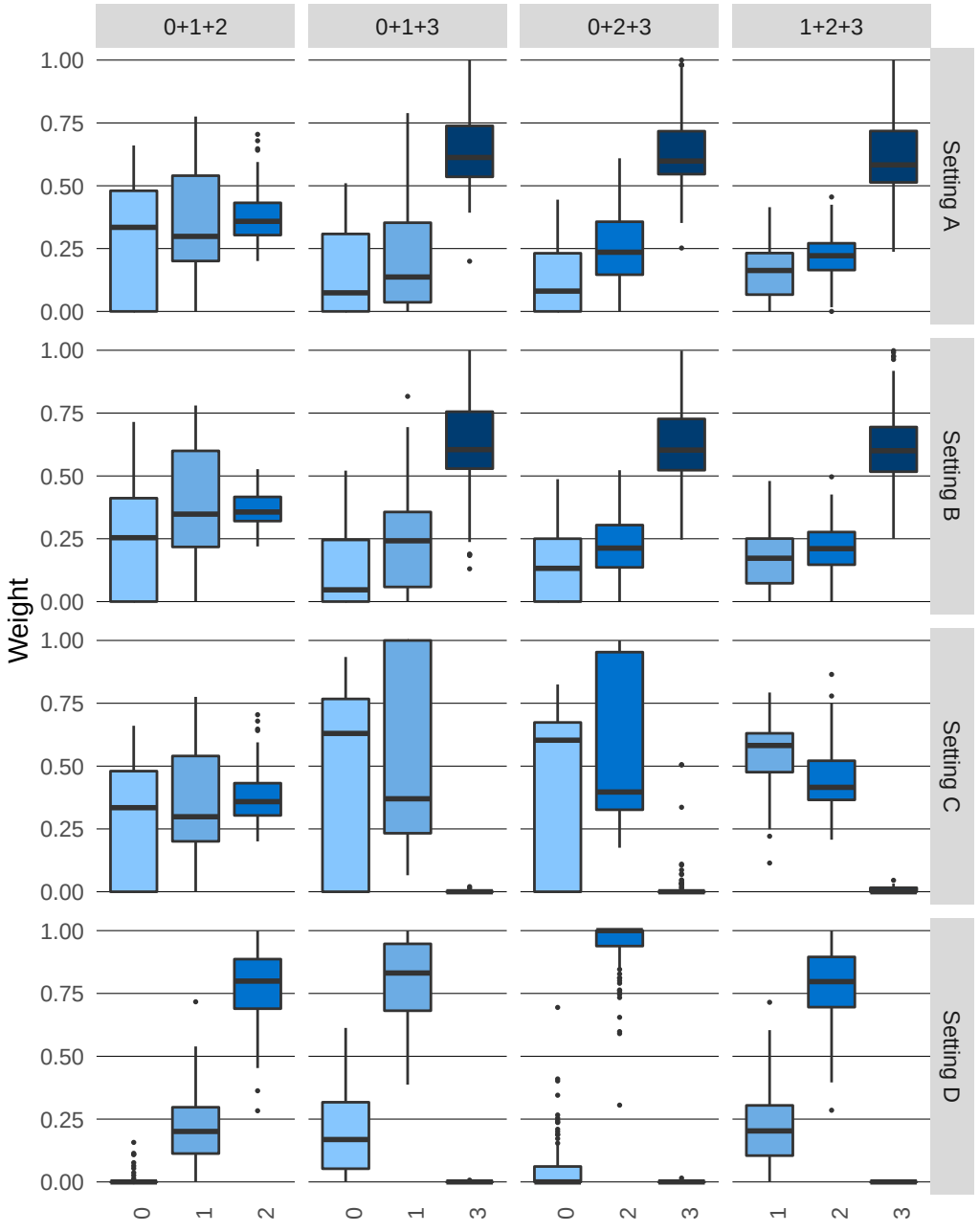


FIGURE C.8: Weights assigned to each PSM for each subset of datasets by the outcome-guided integration method for the simulation study where the response for each observation is given by its true cluster label.

C.2 MULTIPLATFORM ANALYSIS OF TEN CANCER TYPES

We present here some additional figures and results for the multiplatform analysis of ten cancer types of Section 4.4.

First, we give more details about the variable selection step performed using the methodology developed in Chapter 2. Then, we assess the convergence of the MCMC chains on the full and reduced datasets. In the last part of this section, we present additional figures for the unsupervised and outcome-guided integration of four PSMs described in Chapter 4 and repeat the entire analysis for each of the reduced datasets obtained via variable selection.

C.2.1 *Variable selection*

In Table C.1 are reported the number of variables measured in each layer and the number of selected variables using separate EN on each layer as described in Chapter 2.

Dataset	**Full dataset**	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1$
Protein expression	131	131	131	124
mRNA expression	6000	1893	568	258
Methylation	2043	1439	623	322
DNA copy number	84	84	84	84
miRNA expression	51	51	51	50

TABLE C.1: Number of selected variables in each dataset for different values of the EN parameter α.

The full mRNA dataset is too large to be used as input to MDI and for this reason it is only used for the integration of the reduced datasets obtained via variable selection with values of α of 0.5 and 1.

C.2.2 *MCMC convergence assessment*

We run five MCMC chains for 50,000 iterations, with a burn-in period of 25,000 iterations and thinning of 5. For each set of five chains, we check the Vats-Knudson $\hat{R}$ (Vats and Knudson, 2018) with parameters $\epsilon = 0.1$ and $\alpha = 0.1$ to assess the convergence of the mass parameter. The PSMs obtained for the five chains are summarised into one by taking the average.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.87	0.80	0.76	0.69
Chain 2	1	0.91	0.87	0.73
Chain 3		1	0.82	0.68
Chain 4			1	0.83

TABLE C.2: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

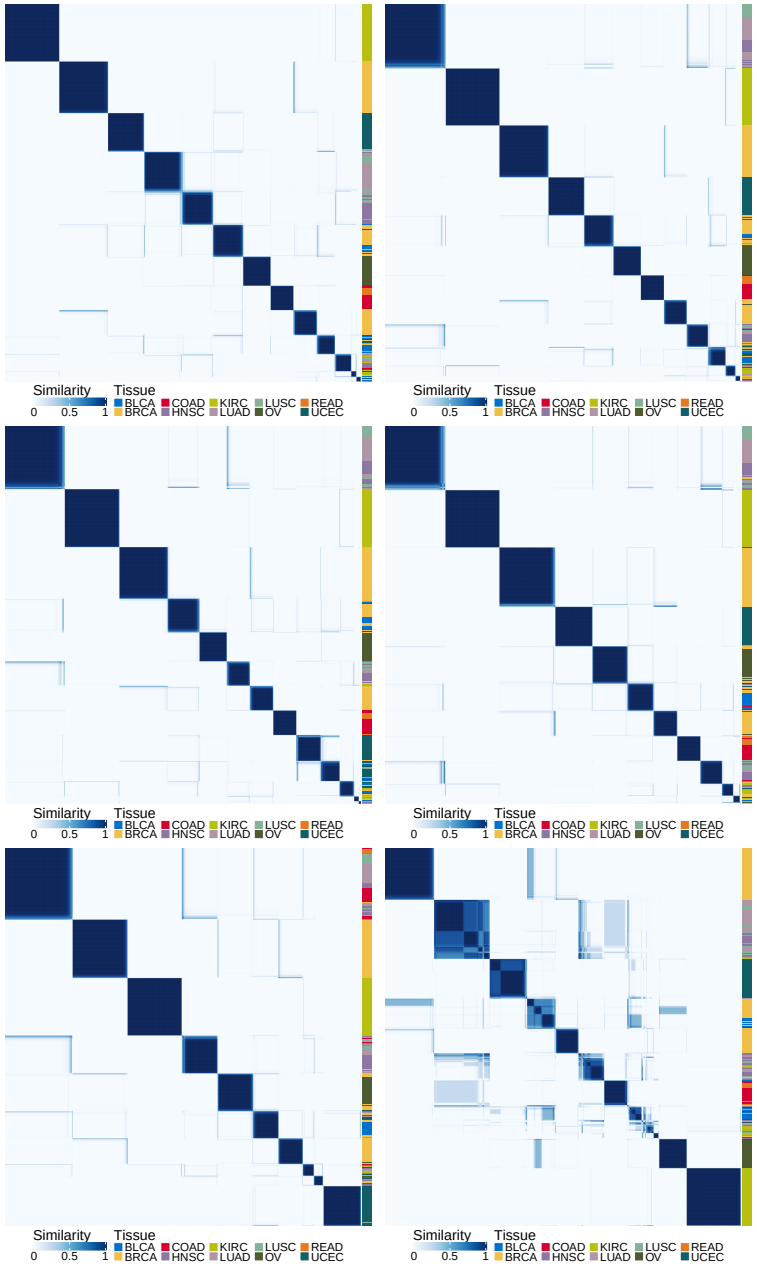


FIGURE C.9: Five PSMs of the protein expression data and their average (bottom right). $\lambda = 0, \alpha = 0.1, 0.5$. On the right of each PSM is indicated the tissue of origin of each tumour sample.

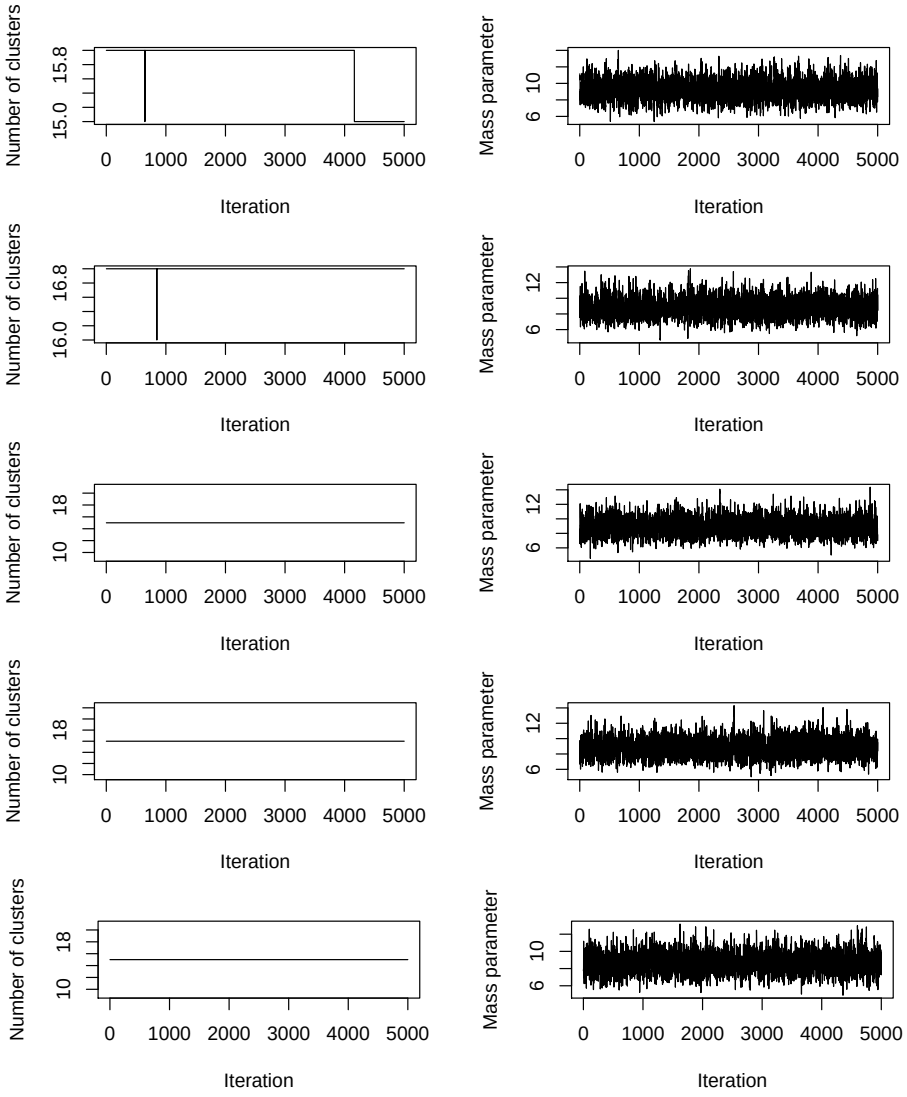


FIGURE C.10: MCMC convergence assessment, protein expression data. $\lambda = 0$, $\alpha = 0.1, 0.5$.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.75	0.72	0.89	0.79
Chain 2	1	0.82	0.80	0.92
Chain 3		1	0.76	0.81
Chain 4			1	0.80

TABLE C.3: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

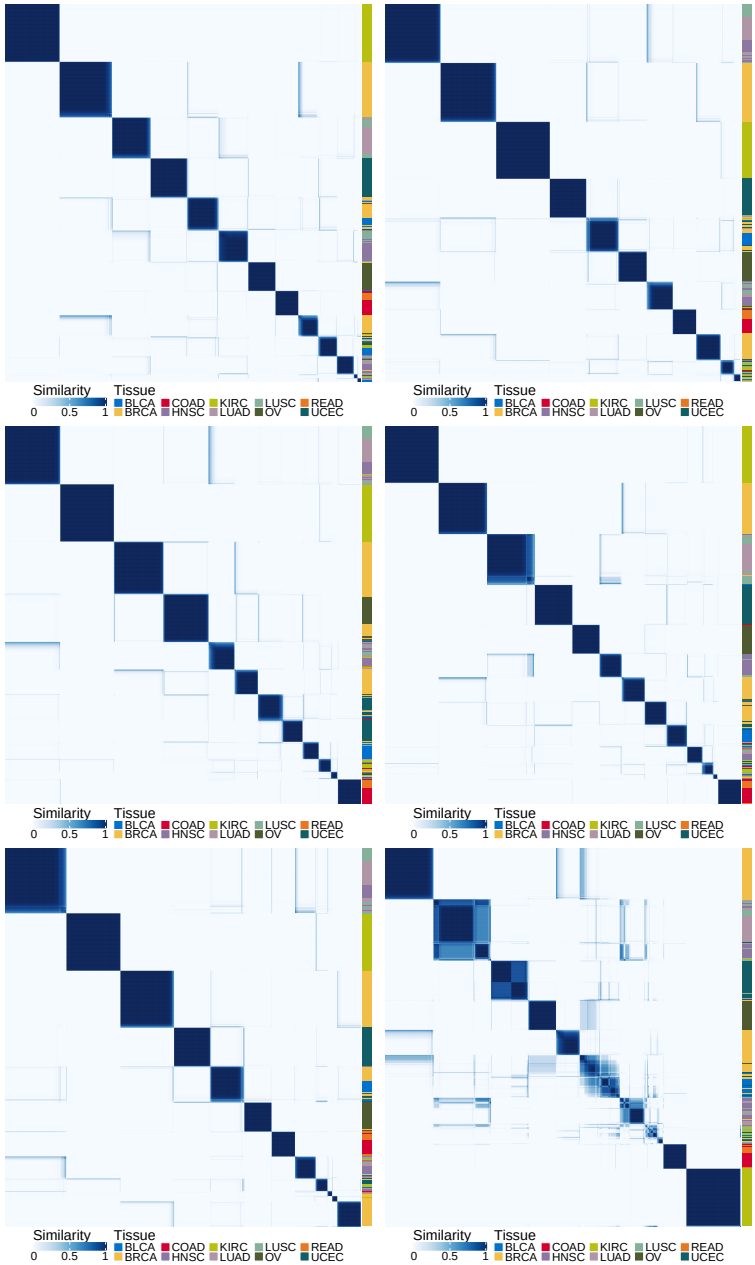


FIGURE C.11: Five PSMs of the protein expression data and their average (bottom right). $\alpha = 1$. On the right of each PSM is indicated the tissue of origin of each tumour sample.

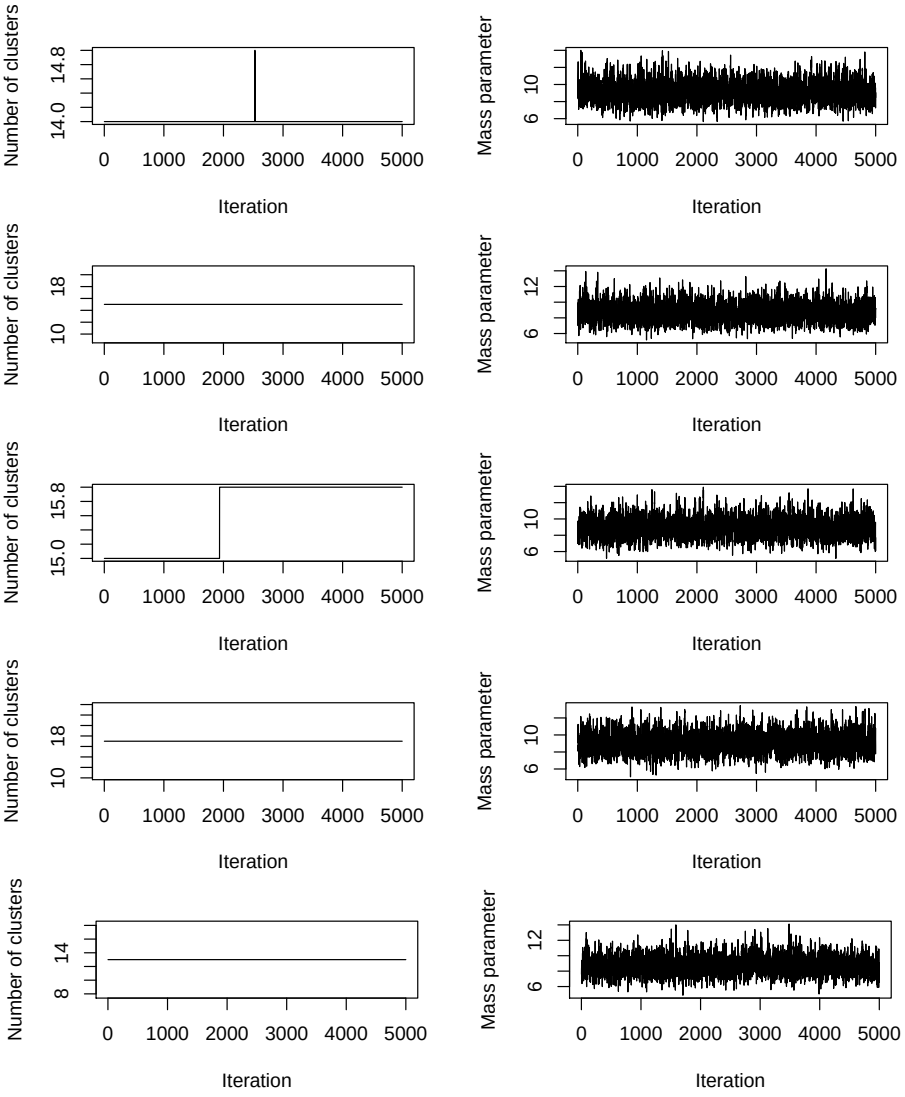


FIGURE C.12: MCMC convergence assessment, protein expression data. $\alpha = 1$.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.46	0.43	0.48	0.42
Chain 2	1	0.42	0.52	0.42
Chain 3		1	0.43	0.41
Chain 4			1	0.49

TABLE C.4: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

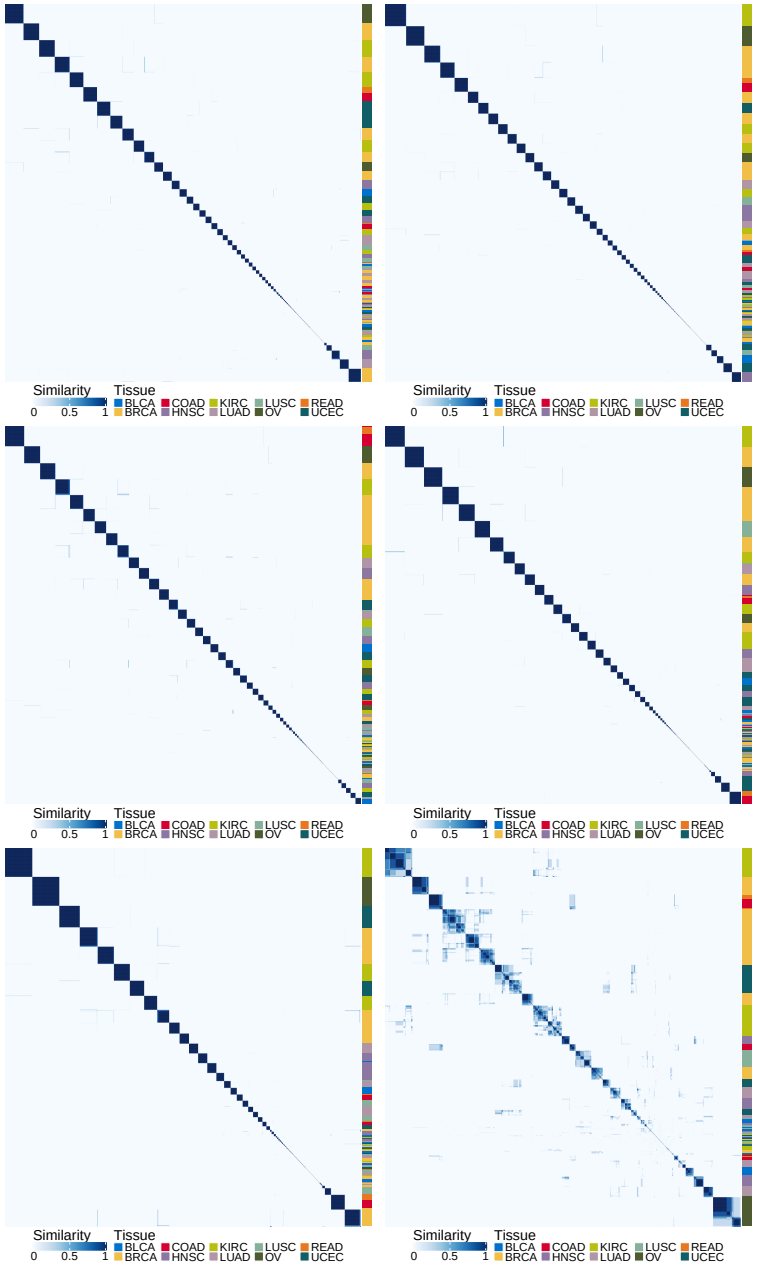


FIGURE C.13: Five PSMs of the mRNA expression data and their average (bottom right). $\alpha = 0.5$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

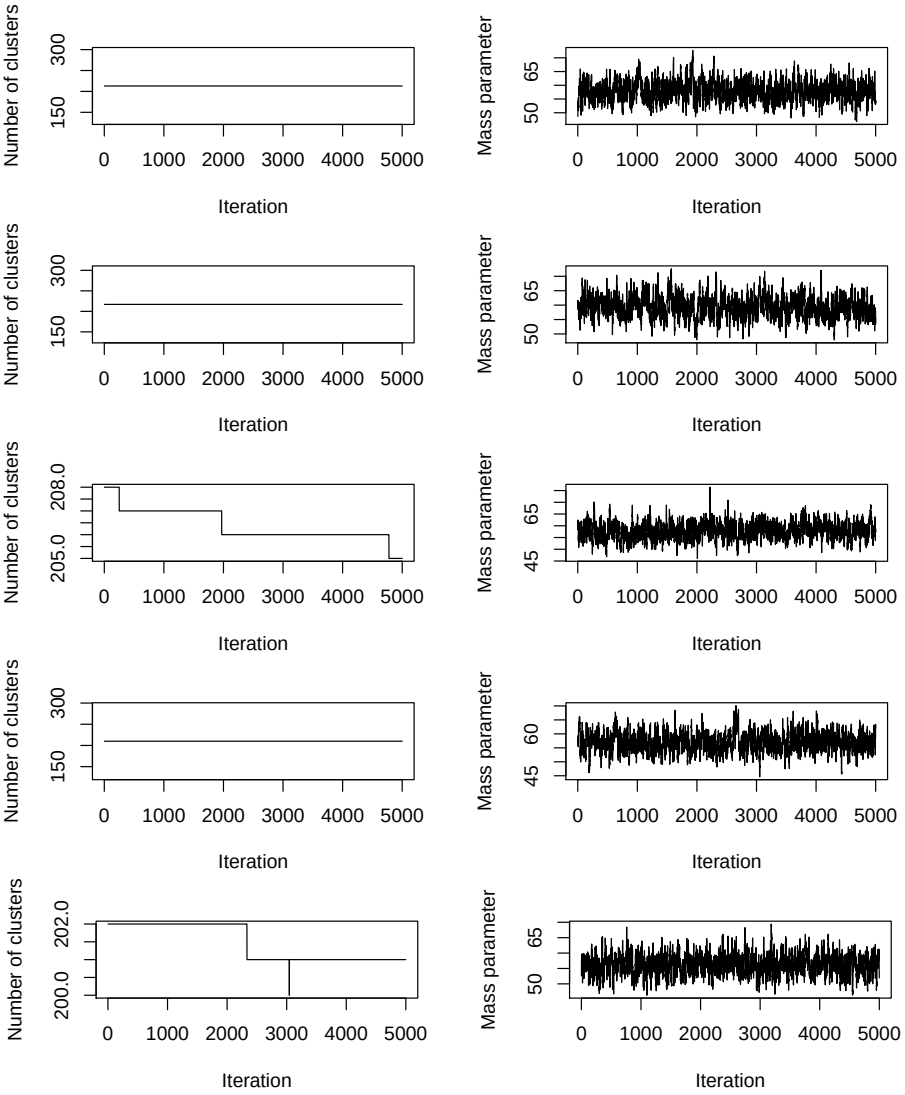


FIGURE C.14: MCMC convergence assessment, mRNA expression data. $\alpha = 0.5$.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.63	0.55	0.61	0.52
Chain 2	1	0.58	0.64	0.63
Chain 3		1	0.59	0.58
Chain 4			1	0.57

TABLE C.5: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

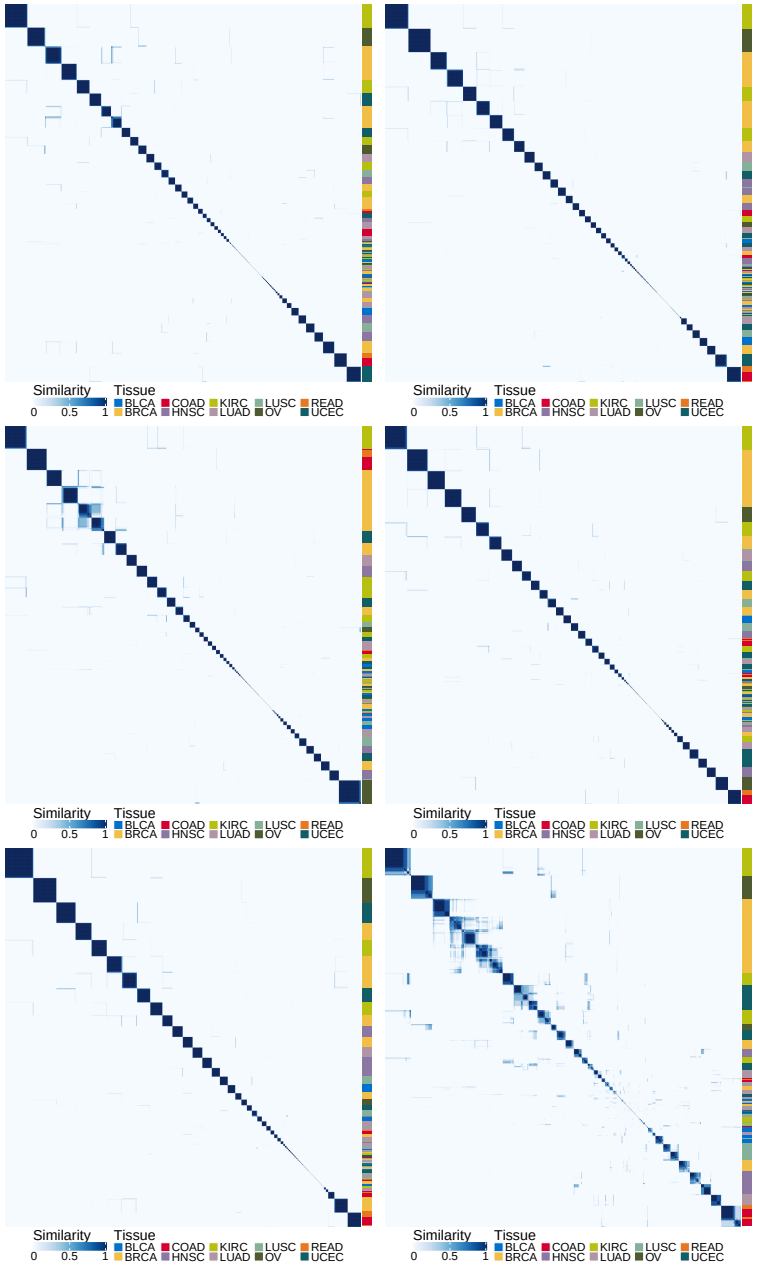


FIGURE C.15: Five PSMs of the mRNA expression data and their average (bottom right). $\alpha = 1$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

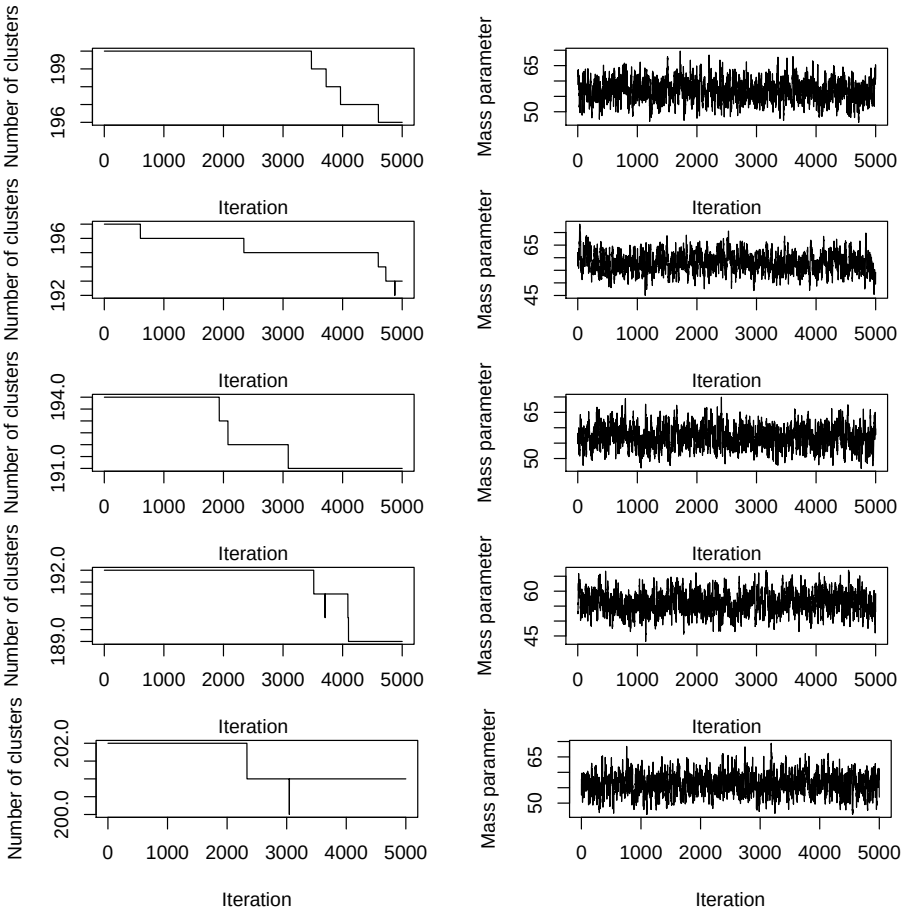


FIGURE C.16: MCMC convergence assessment, mRNA expression data. $\alpha = 1$.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.60	0.57	0.53	0.79
Chain 2	1	0.67	0.63	0.64
Chain 3		1	0.59	0.66
Chain 4			1	0.54

TABLE C.6: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

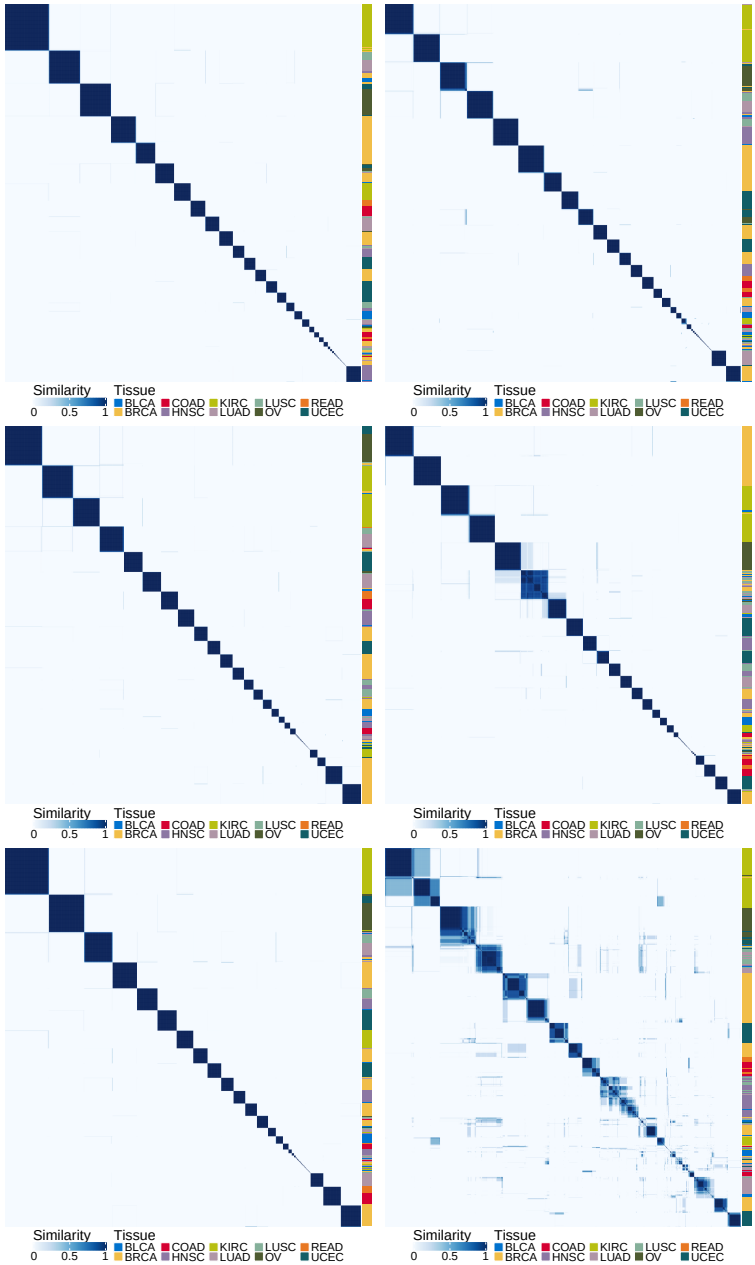


FIGURE C.17: Five PSMs of the methylation data and their average (bottom right). $\lambda = 0$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

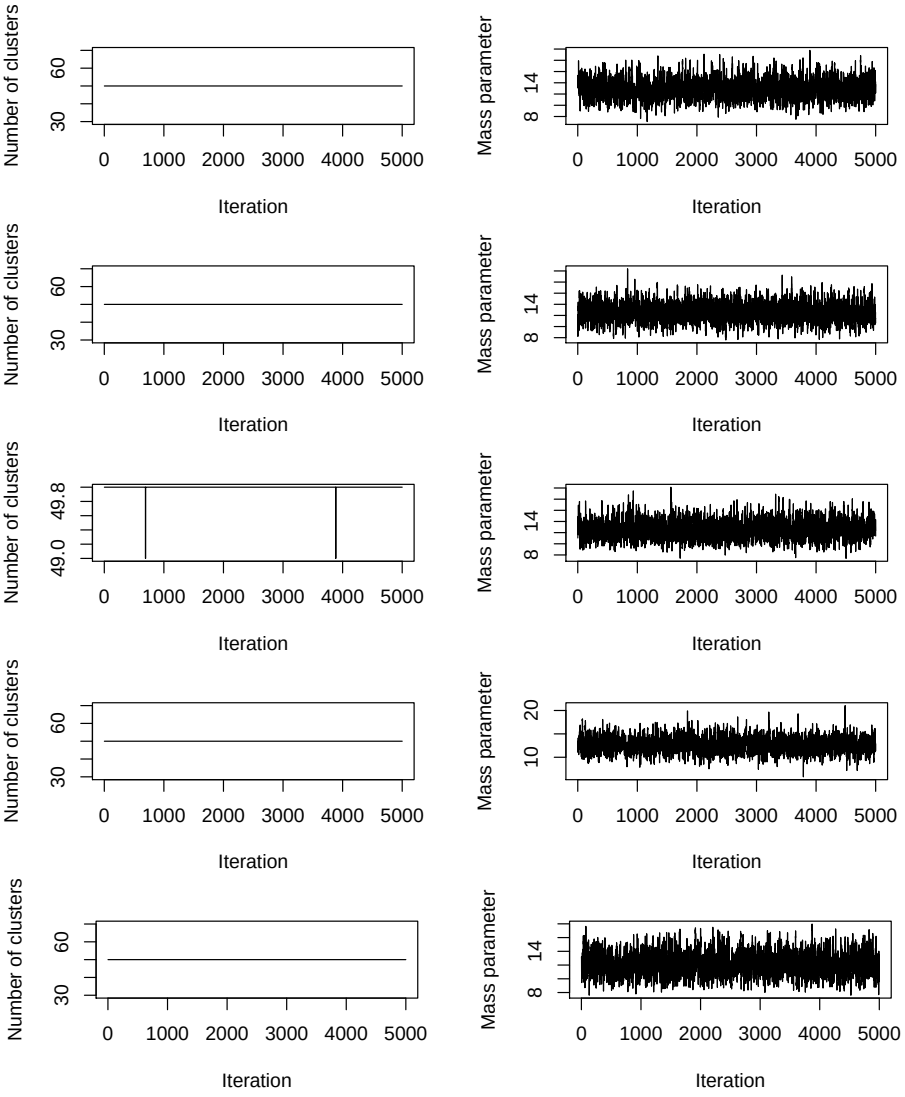


FIGURE C.18: MCMC convergence assessment, methylation data. $\lambda = 0$.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.55	0.53	0.61	0.64
Chain 2	1	0.48	0.57	0.49
Chain 3		1	0.71	0.55
Chain 4			1	0.57

TABLE C.7: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

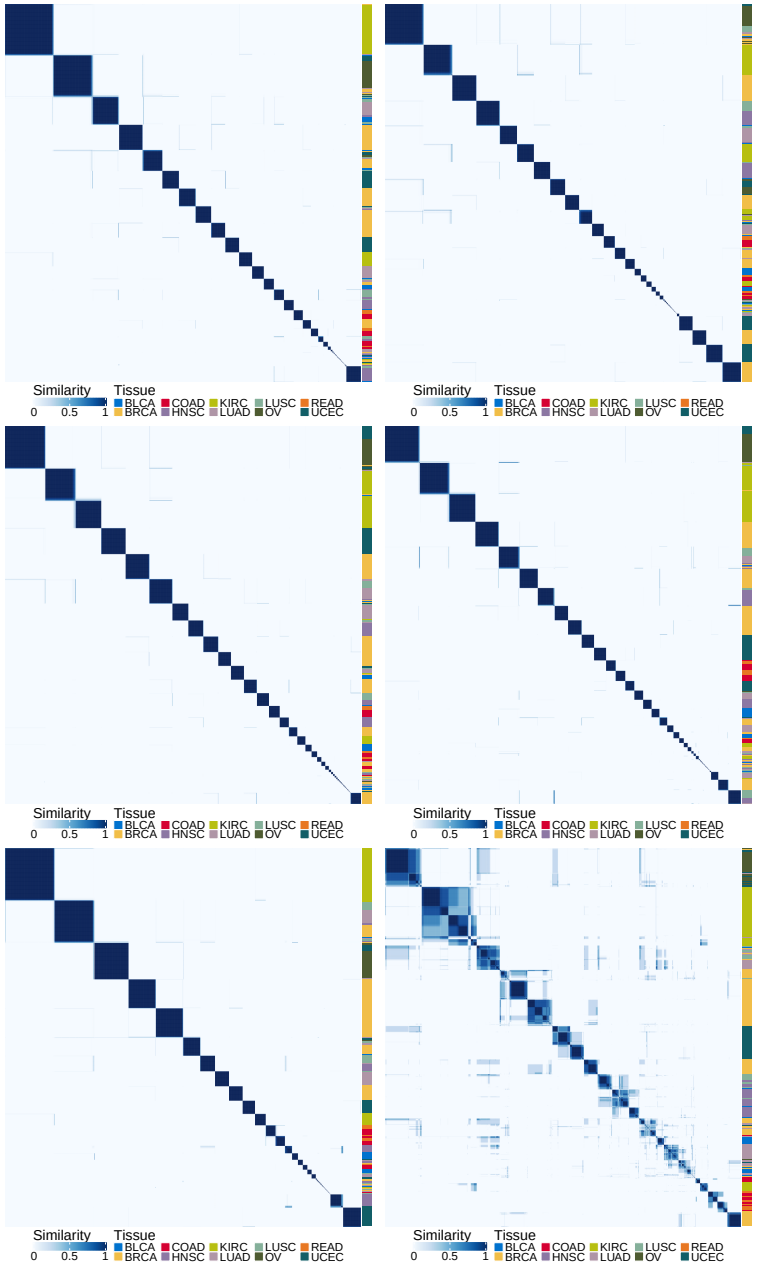


FIGURE C.19: Five PSMs of the methylation data and their average (bottom right). $\alpha = 0.1$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

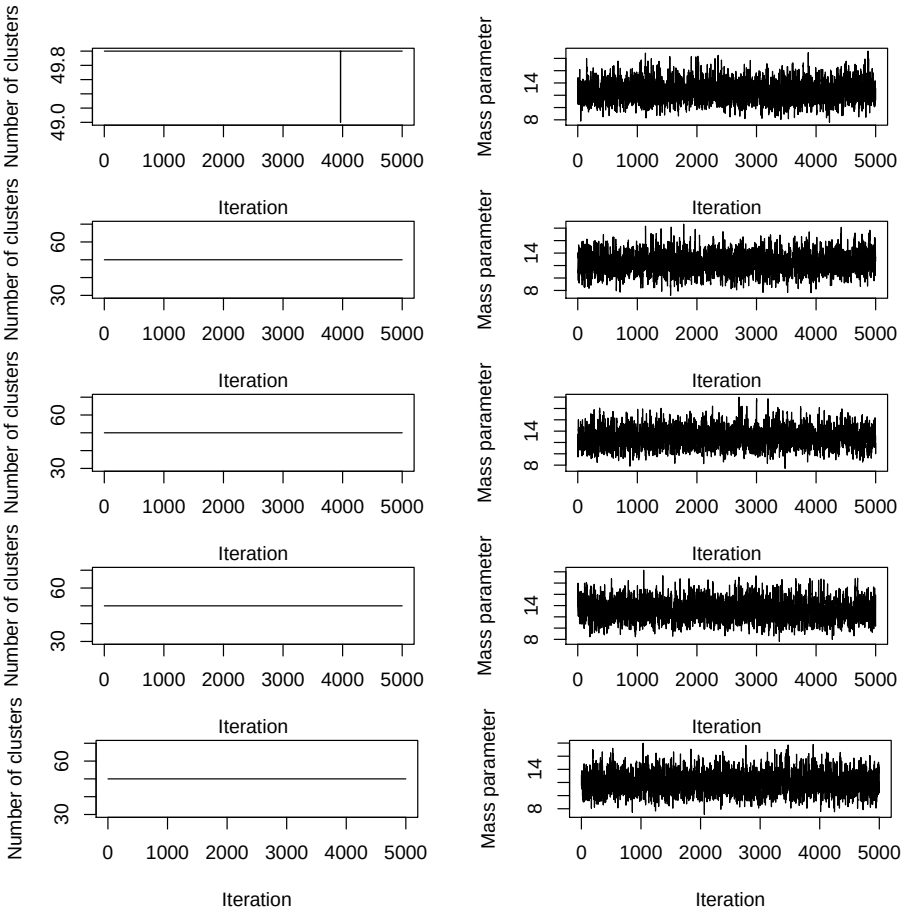


FIGURE C.20: MCMC convergence assessment, methylation data. $\alpha = 0.1$

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.79	0.74	0.74	0.78
Chain 2	1	0.81	0.73	0.82
Chain 3		1	0.70	0.77
Chain 4			1	0.75

TABLE C.8: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

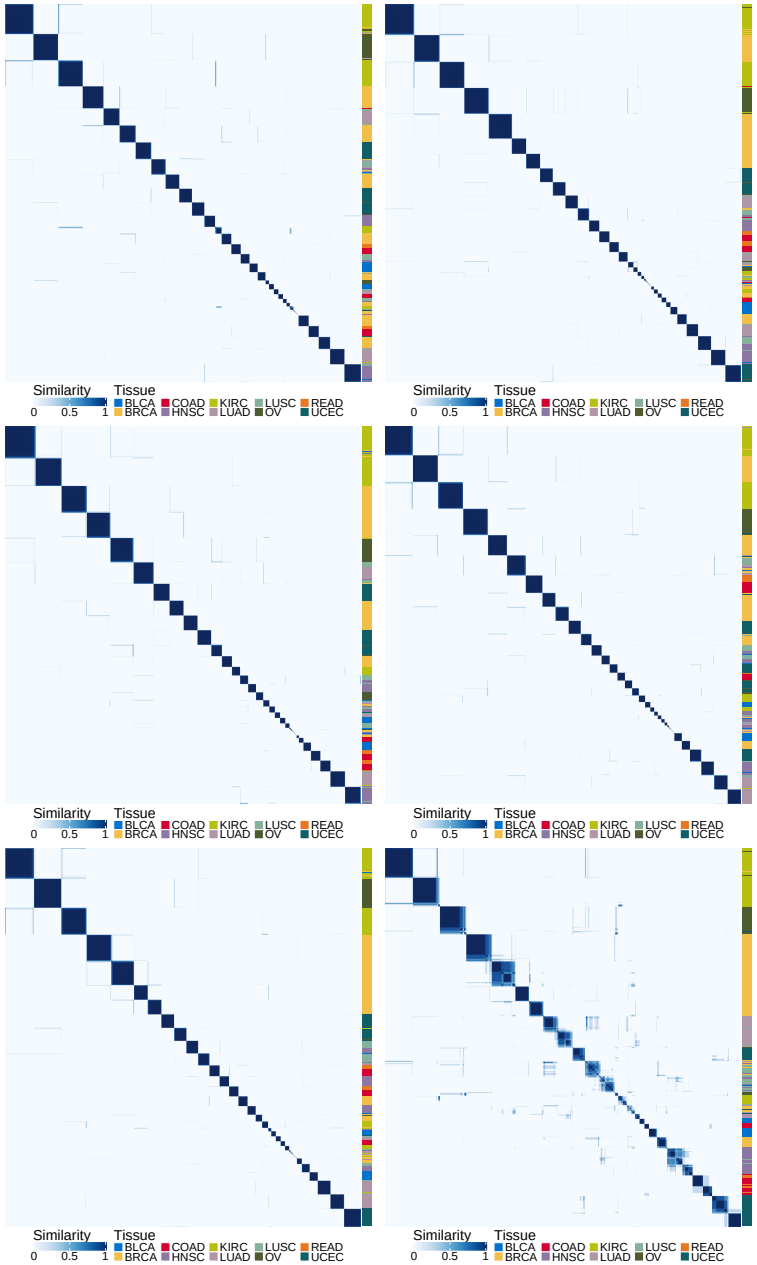


FIGURE C.21: Five PSMs of the methylation data and their average (bottom right). $\alpha = 0.5$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

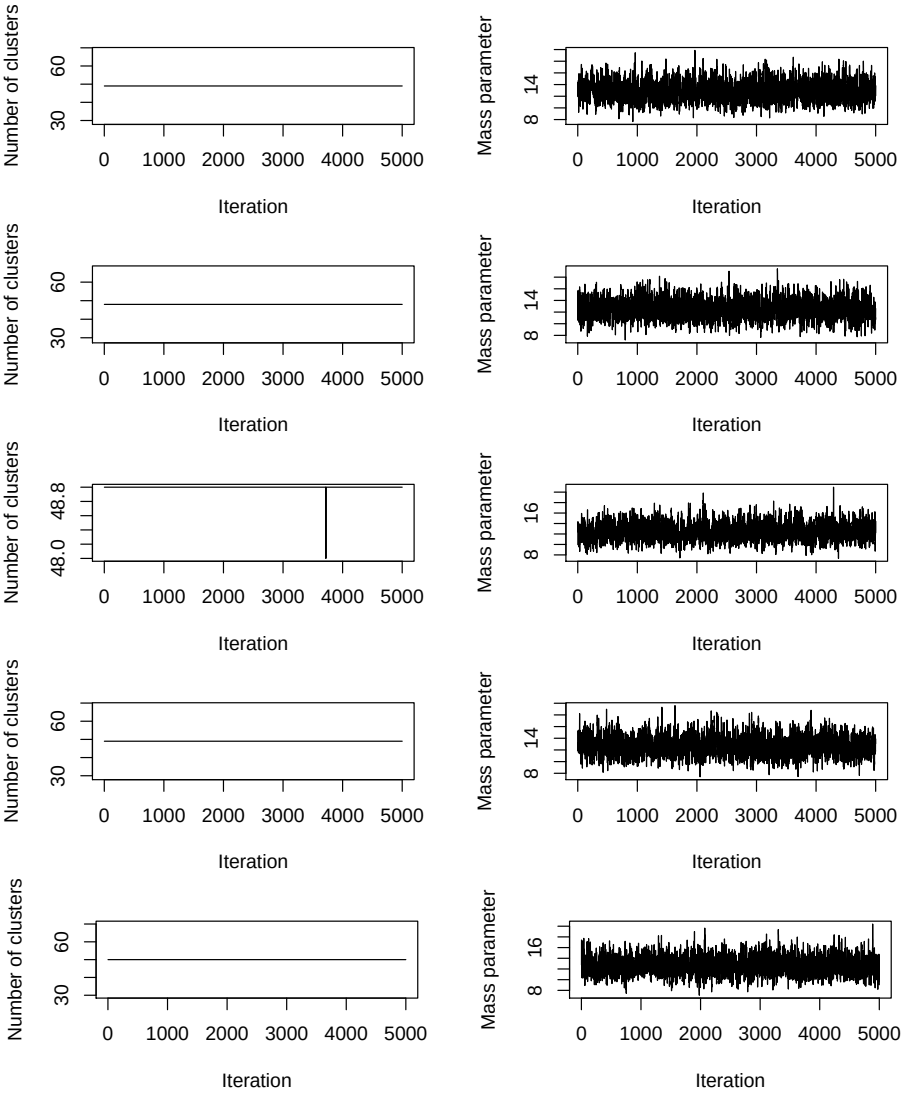


FIGURE C.22: MCMC convergence assessment, methylation data.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.81	0.70	0.83	0.78
Chain 2	1	0.61	0.76	0.67
Chain 3		1	0.67	0.60
Chain 4			1	0.76

TABLE C.9: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

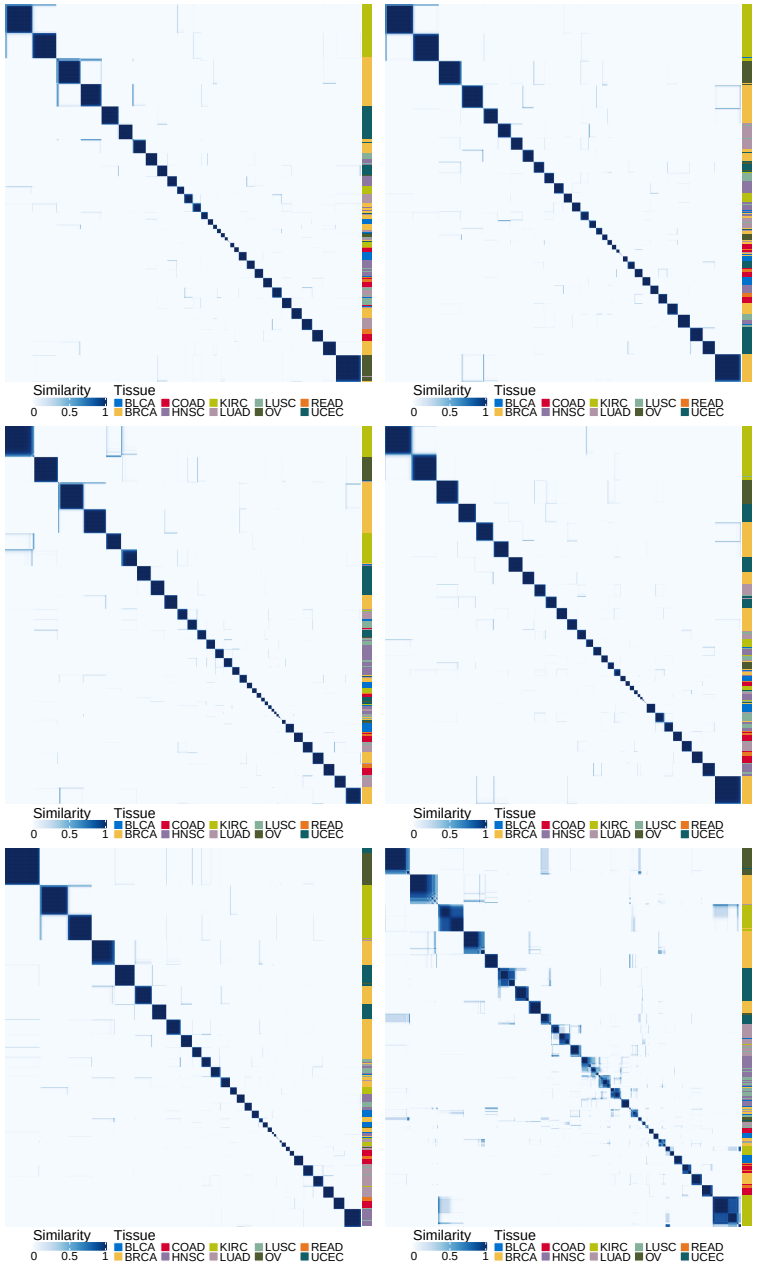


FIGURE C.23: Five PSMs of the methylation data and their average (bottom right). $\alpha = 1$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

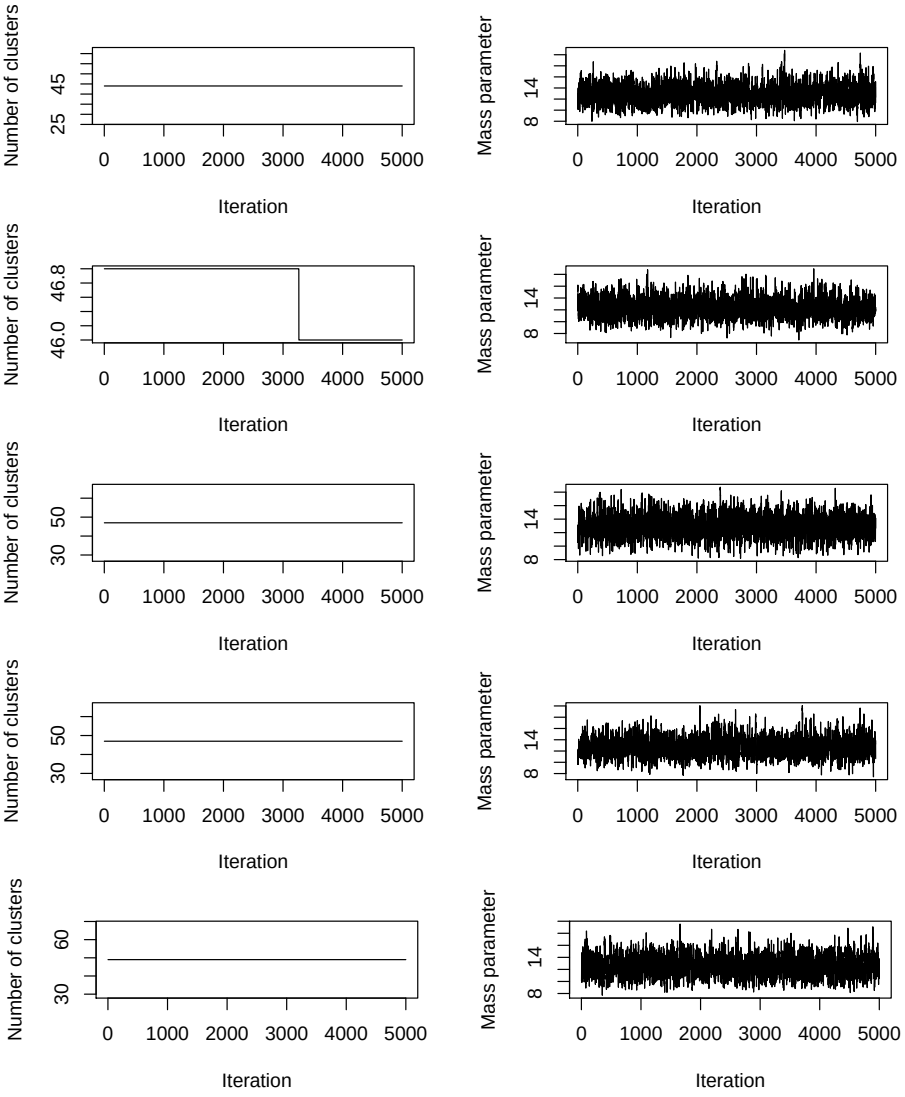


FIGURE C.24: MCMC convergence assessment, methylation data.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.39	0.45	0.35	0.35
Chain 2	1	0.58	0.60	0.62
Chain 3		1	0.59	0.57
Chain 4			1	0.54

TABLE C.10: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

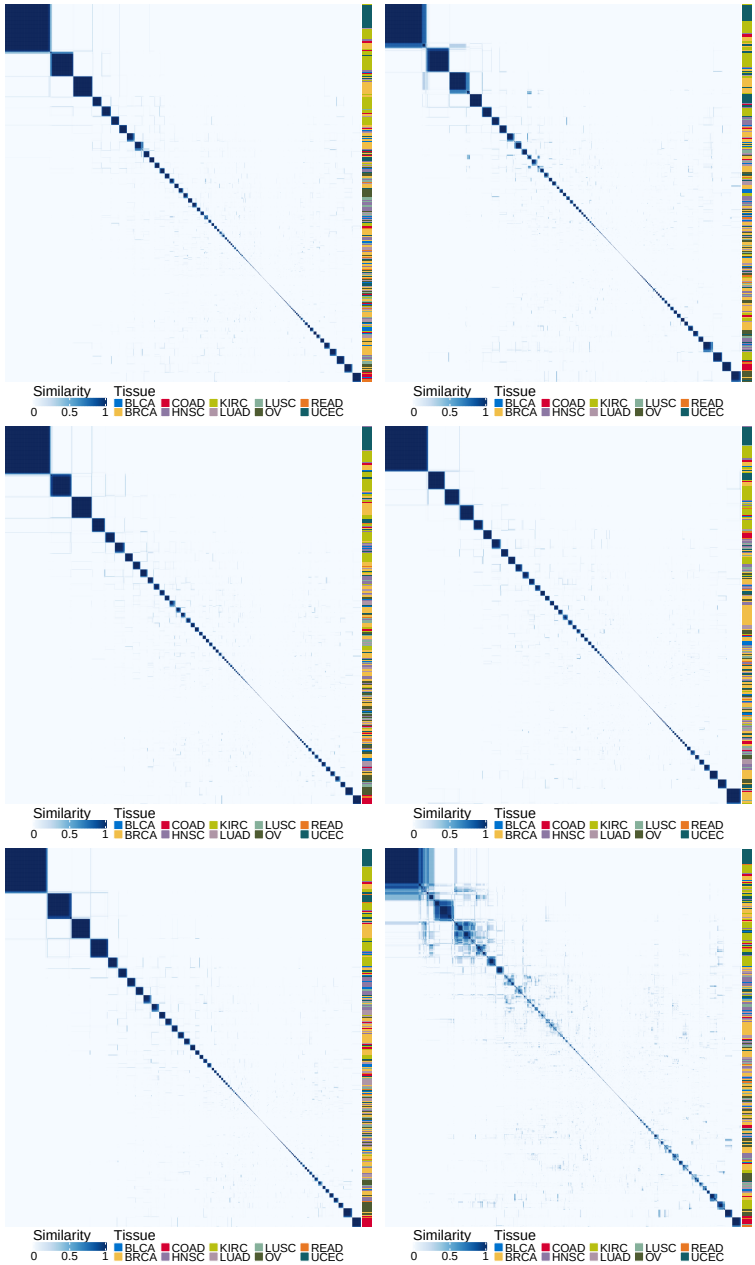


FIGURE C.25: Five PSMs of the DNA copy number data and their average (bottom right).

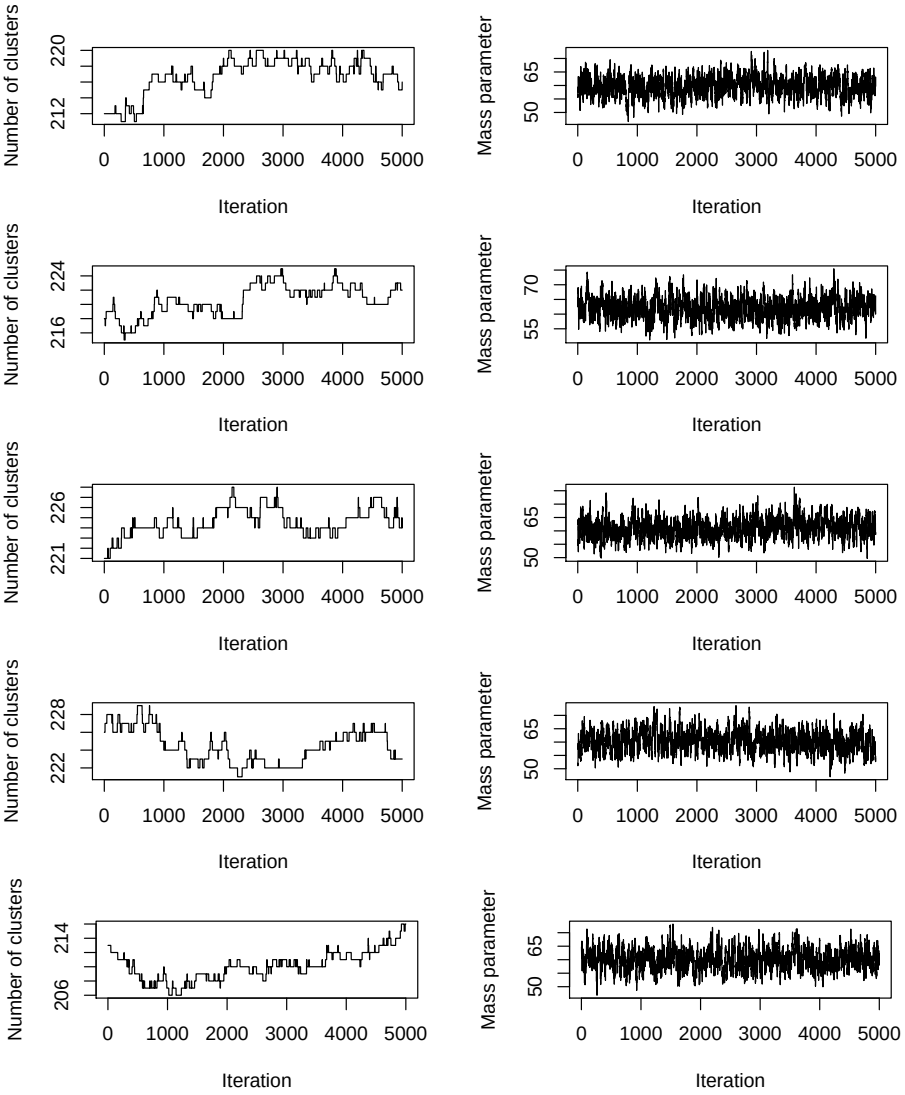


FIGURE C.26: MCMC convergence assessment, DNA copy number data.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.63	0.75	0.59	0.64
Chain 2	1	0.74	0.82	0.82
Chain 3		1	0.68	0.69
Chain 4			1	0.75

TABLE C.11: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

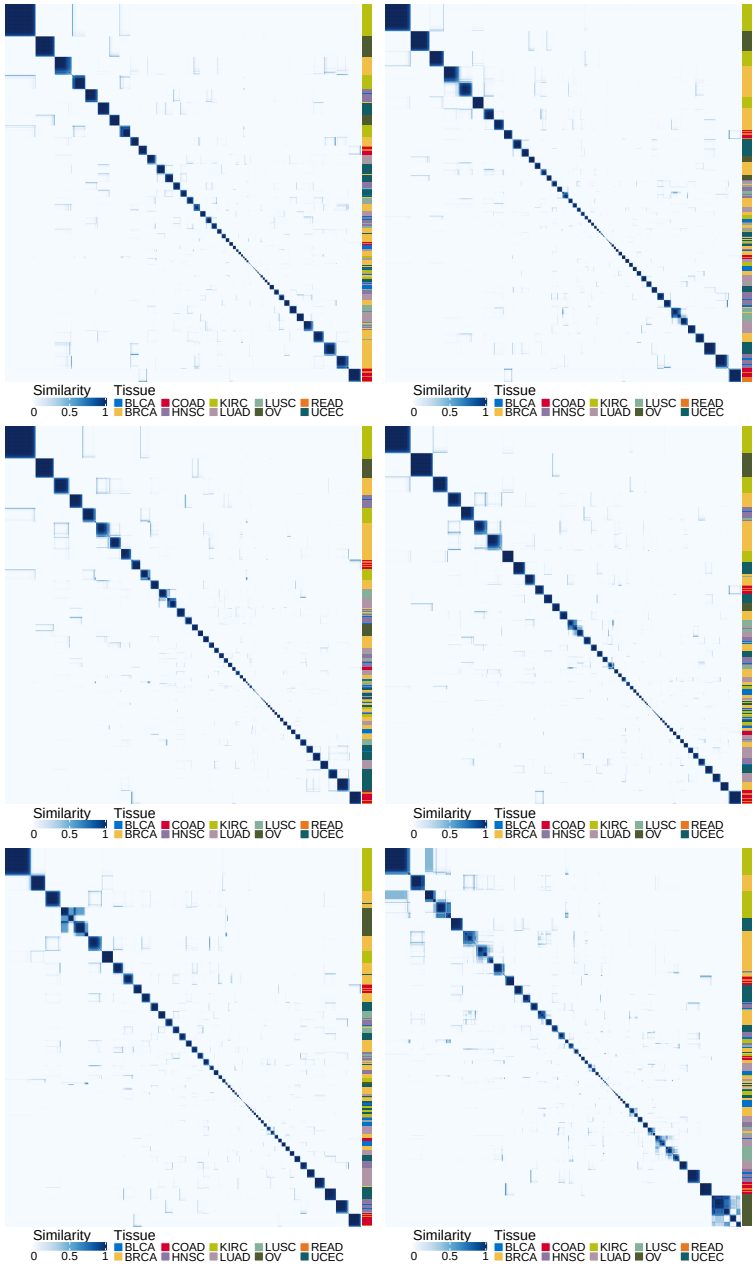


FIGURE C.27: Five PSMs of the miRNA data and their average (bottom right). $\lambda = 0$, $\alpha = 0.1, 0.5$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

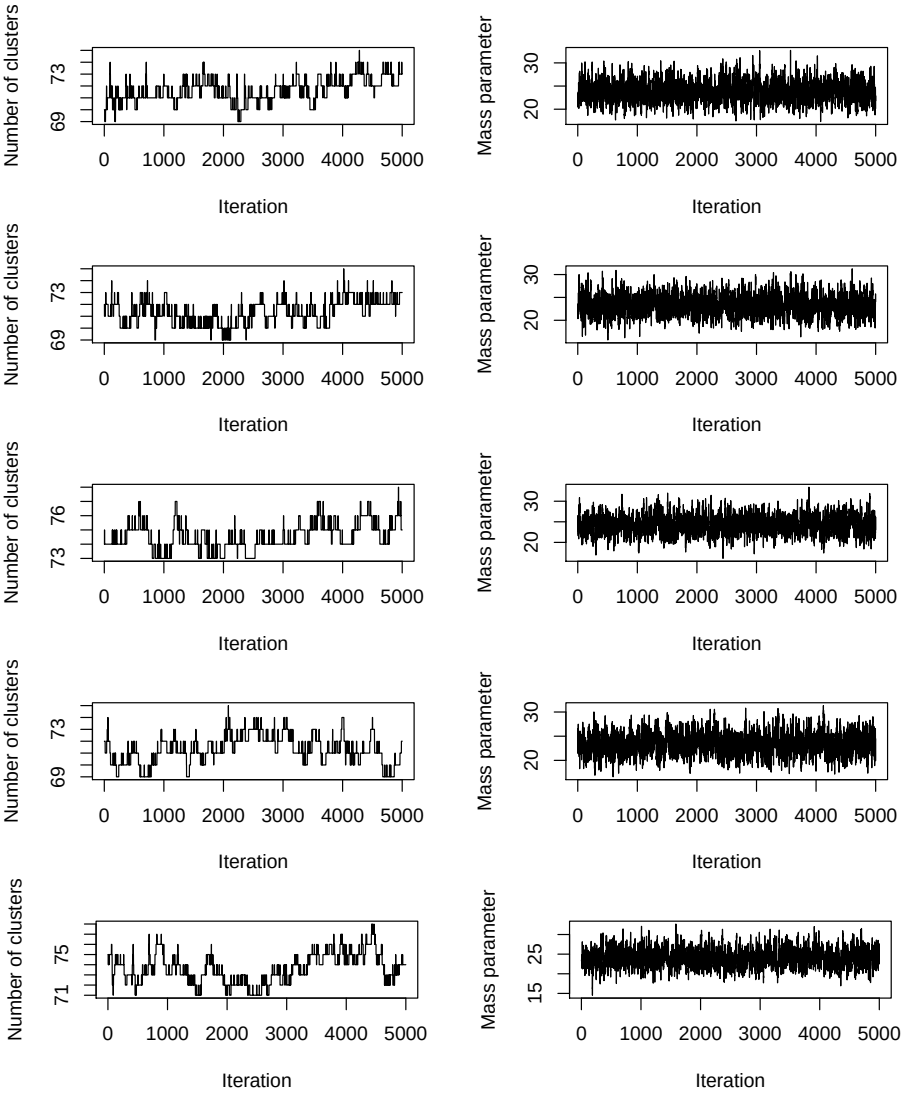


FIGURE C.28: MCMC convergence assessment, miRNA. $\lambda = 0, \alpha = 0.1, 0.5$.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.66	0.72	0.70	0.57
Chain 2	1	0.59	0.76	0.69
Chain 3		1	0.55	0.56
Chain 4			1	0.72

TABLE C.12: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

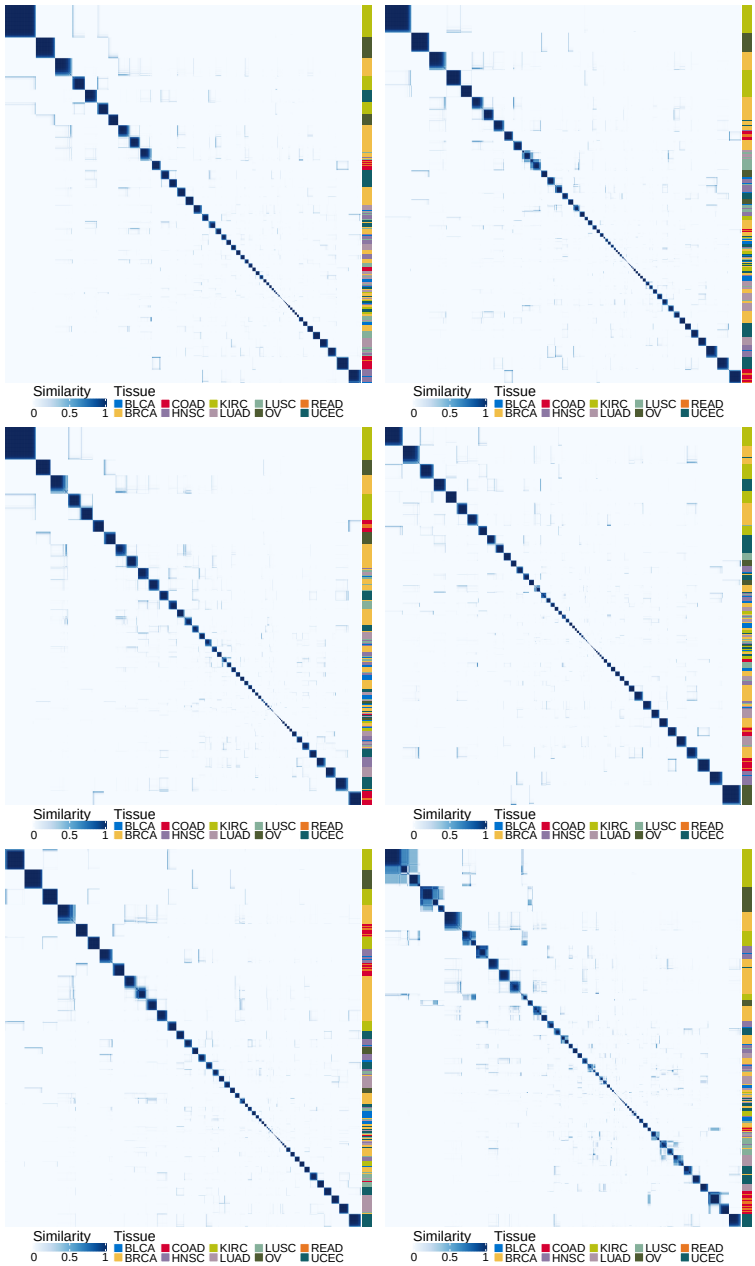


FIGURE C.29: Five PSMs of the miRNA data and their average (bottom right). $\alpha = 1$. On the left of each PSM is indicated the tissue of origin of each tumour sample.

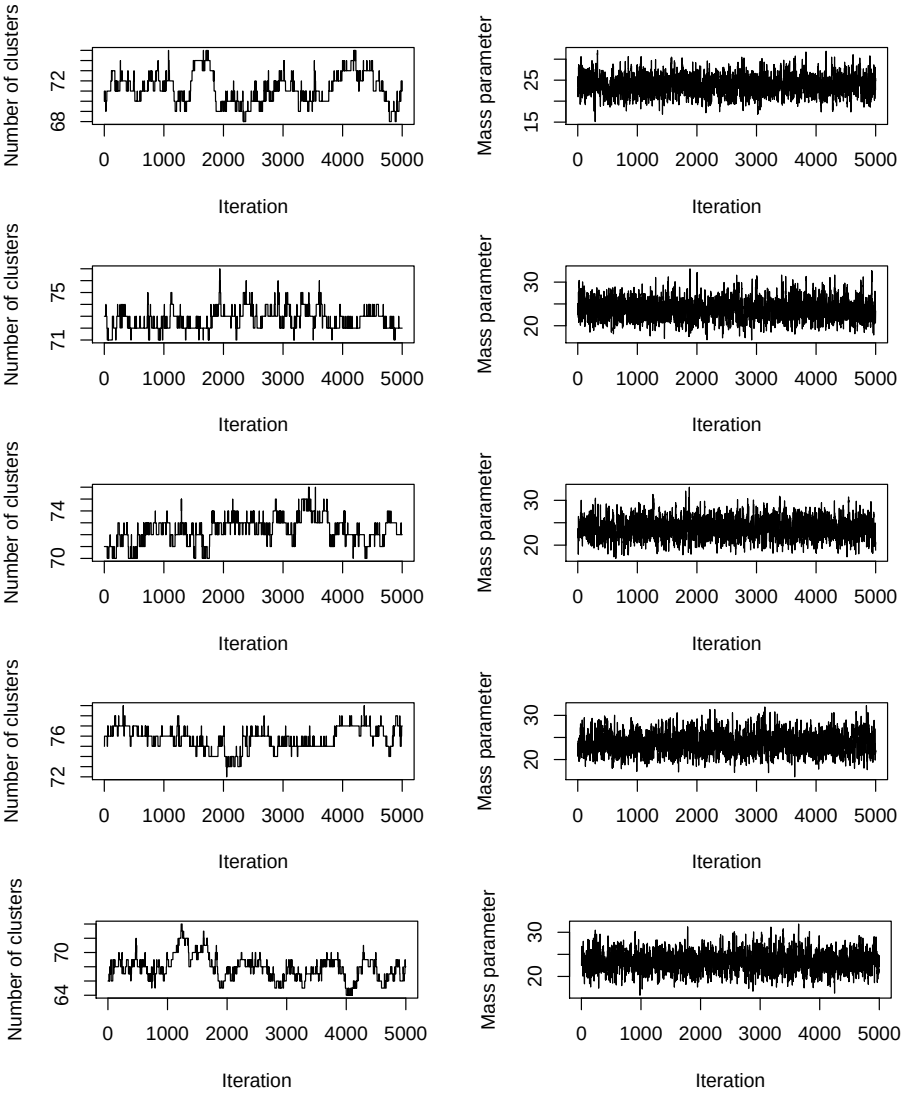


FIGURE C.30: MCMC convergence assessment, miRNA data. $\alpha = 1$.

C.2.3 *Unsupervised integration: additional figures and results*

First, we report some additional figures for the unsupervised integration of Section 4.4, then we repeat the analysis with the reduced datasets obtained as described in Section C.2.1.

Comparison with the clusters identified by Hoadley et al. (2014)

In Figure C.31 are shown the correspondences between the clusters found in the main paper and the clusters identified by Hoadley et al. (2014) using Cluster-Of-Clusters Analysis (COCA).

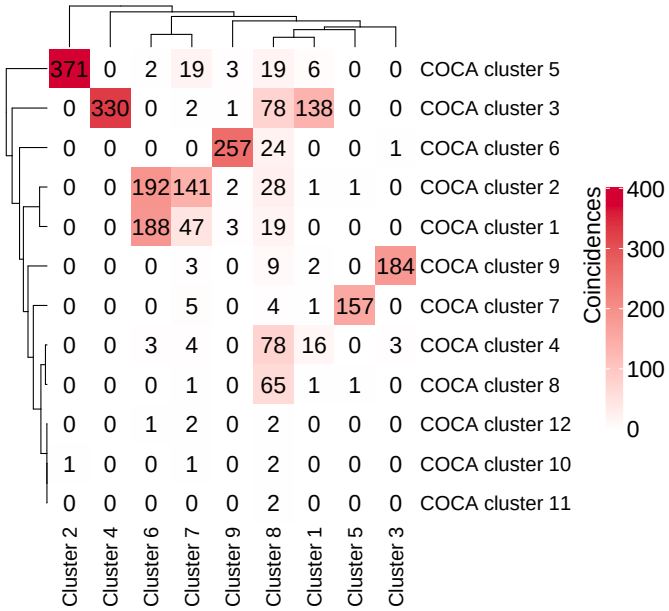


FIGURE C.31: Comparison between the clusters found combining the PSMs of each layer using multiple kernel learning and those identified by Hoadley et al. (2014) using COCA.

Clustering structure in the data

Figures C.32, C.33, C.33, and C.35 show the four data layers where the rows have been sorted by final cluster.

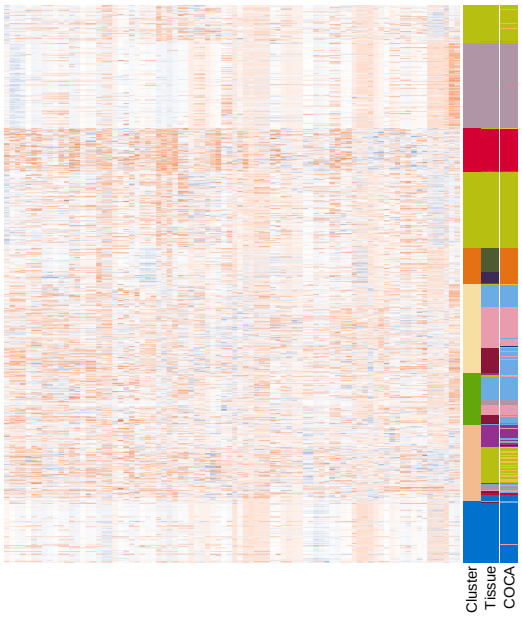


FIGURE C.32: Left: copy number data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

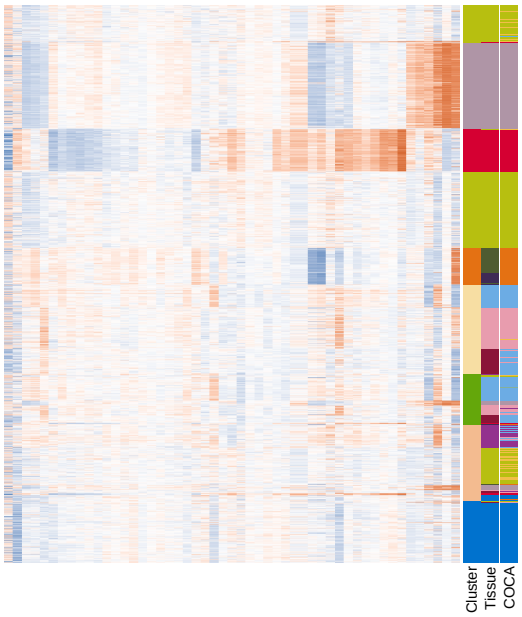


FIGURE C.33: Left: miRNA expression data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

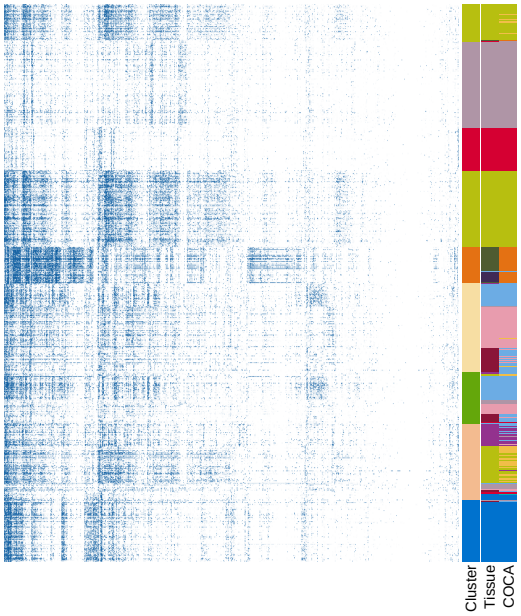


FIGURE C.34: Left: methylation data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

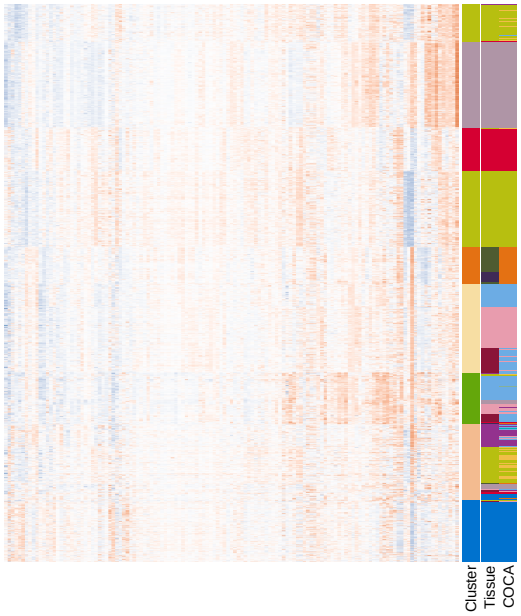
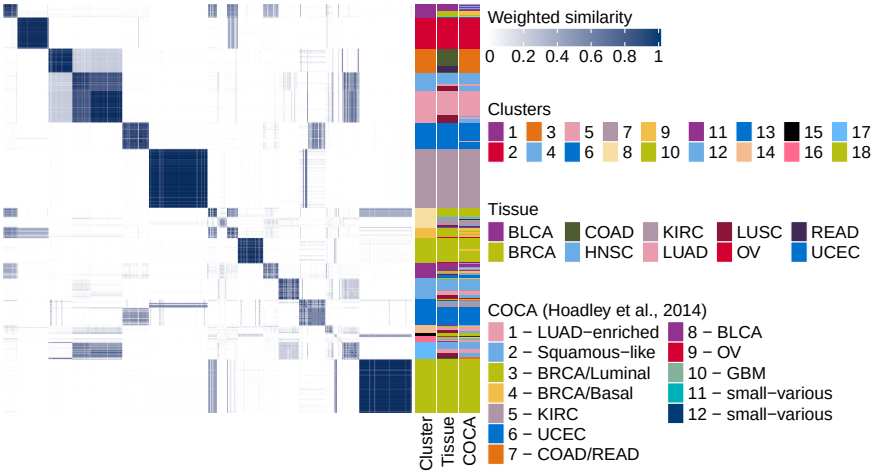


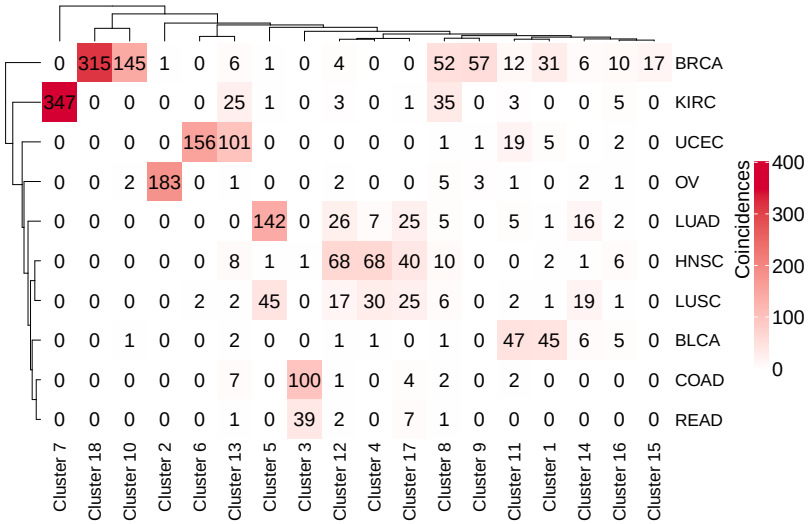
FIGURE C.35: Left: protein expression data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

Unsupervised integration after variable selection, $\alpha = 0.1$

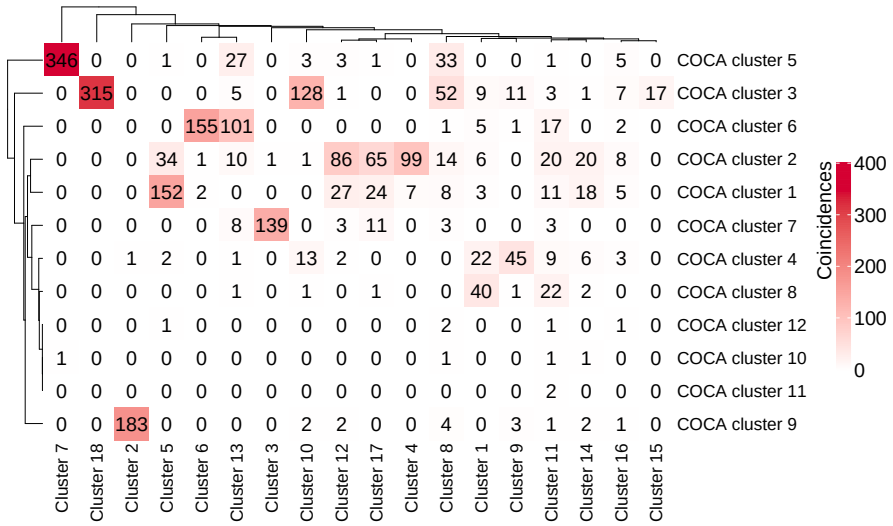


(A) Weighted kernel and clusters.

FIGURE C.36: Unsupervised multiplatform analysis of ten cancer types. Weighted kernel, final clusters, tissues of origin, and COCA clusters.



(A) Comparison to tissue of origin.



(B) Comparison to COCA clusters.

FIGURE C.37: Unsupervised multiplatform analysis of ten cancer types. (A) Coincidence matrix comparing the tissue of origin of the tumour samples to the new clusters. (B) Coincidence matrix comparing the COCA clusters of Hoadley *et al.* to the new clusters.

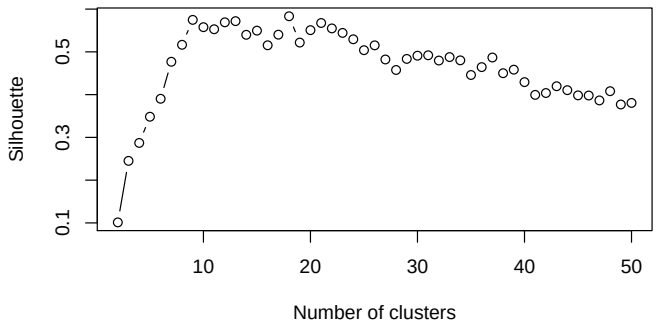


FIGURE C.38: Average silhouette for number of clusters going from 2 to 50.

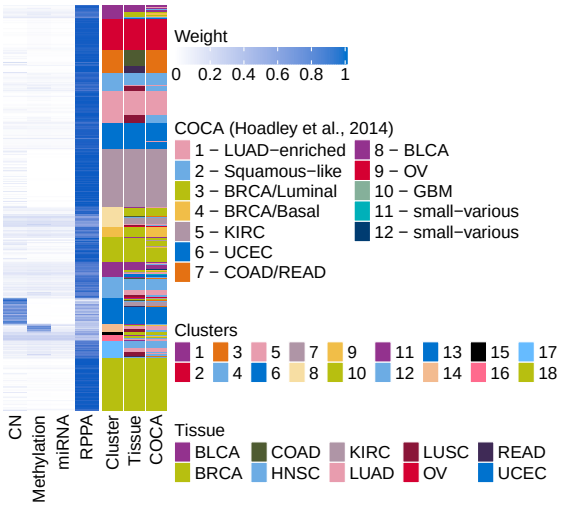
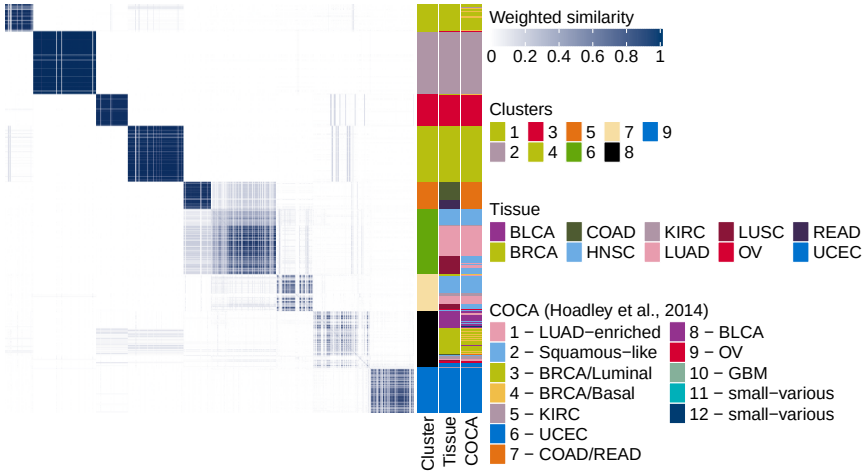
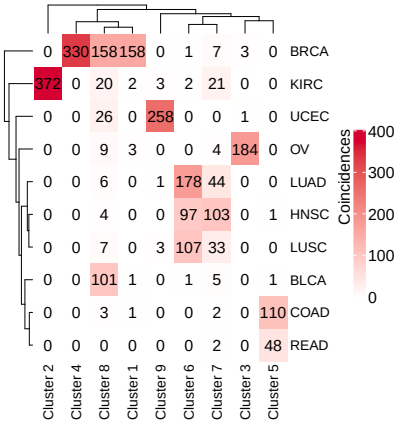


FIGURE C.39: Weights assigned by the multiple kernel *k*-means algorithm to each observations in each layer, where "CN" stands for copy number and "RPPA" for reverse phase protein array. The weights assigned on average to the tumour samples in each layer are: copy number 7.9%, methylation 4.6%, miRNA 3.2%, protein 84.3%.

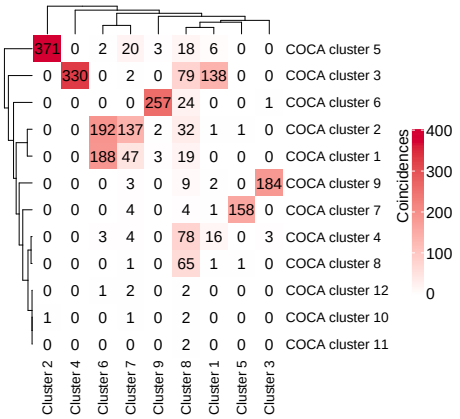
Unsupervised integration after variable selection, $\alpha = 0.5$



(A) Weighted kernel and clusters.



(B) Comparison to tissue of origin.



(C) Comparison to COCA clusters.

FIGURE C.40: Unsupervised multiplatform analysis of ten cancer types. (A) Weighted kernel, final clusters, tissues of origin, and COCA clusters. (B) Coincidence matrix comparing the tissue of origin of the tumour samples to the new clusters. **(c)** Coincidence matrix comparing the COCA clusters of Hoadley *et al.* to the new clusters.

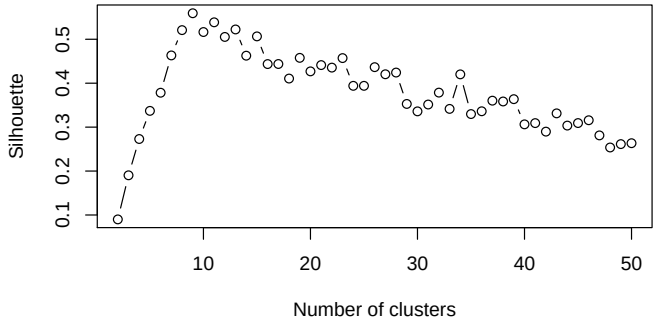


FIGURE C.41: Average silhouette for number of clusters going from 2 to 50.

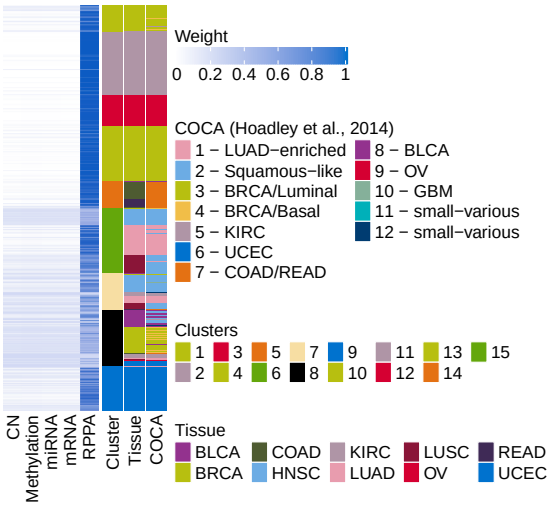
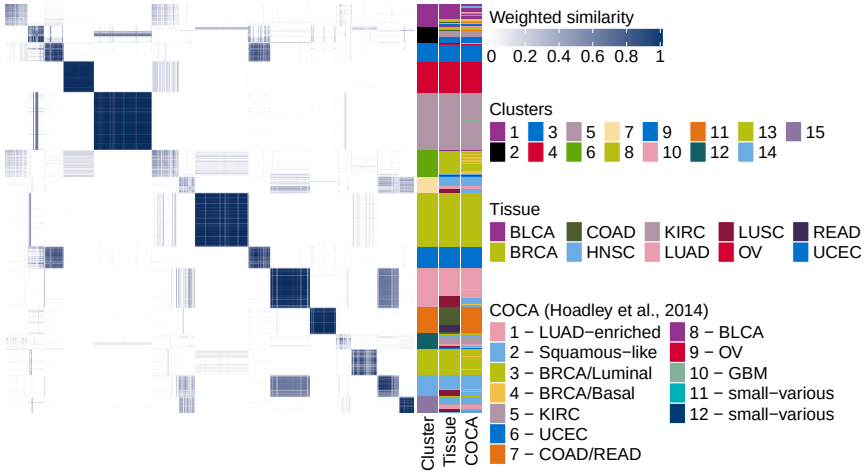
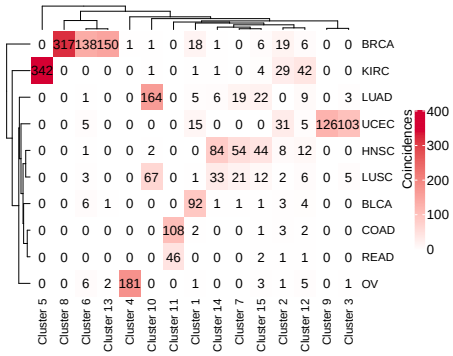


FIGURE C.42: Weights assigned by the multiple kernel *k*-means algorithm to each observations in each layer, where "CN" stands for copy number and "RPPA" for reverse phase protein array. The weights assigned on average to the tumour samples in each layer are: copy number 5.8%, methylation 6%, miRNA 5.9%, mRNA 5.9%, protein 76.4%.

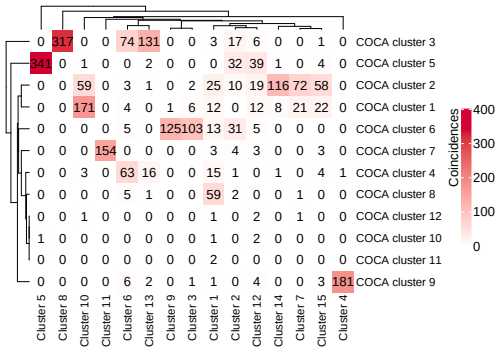
Unsupervised integration after variable selection, $\alpha = 1$



(A) Weighted kernel and clusters.



(B) Comparison to tissue of origin.



(C) Comparison to COCA clusters.

FIGURE C.43: Unsupervised multiplatform analysis of ten cancer types. (A) Weighted kernel, final clusters, tissues of origin, and COCA clusters. (B) Coincidence matrix comparing the tissue of origin of the tumour samples to the new clusters. **(c)** Coincidence matrix comparing the COCA clusters of Hoadley *et al.* to the new clusters.

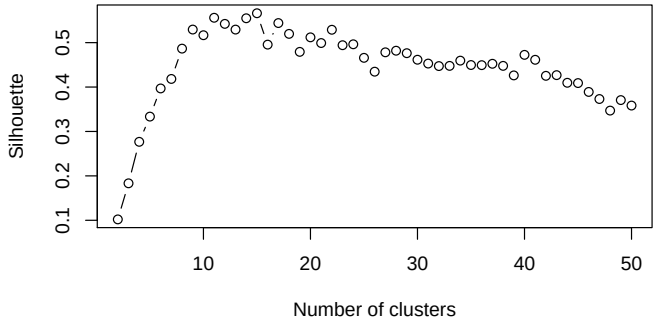


FIGURE C.44: Average silhouette for number of clusters going from 2 to 50.

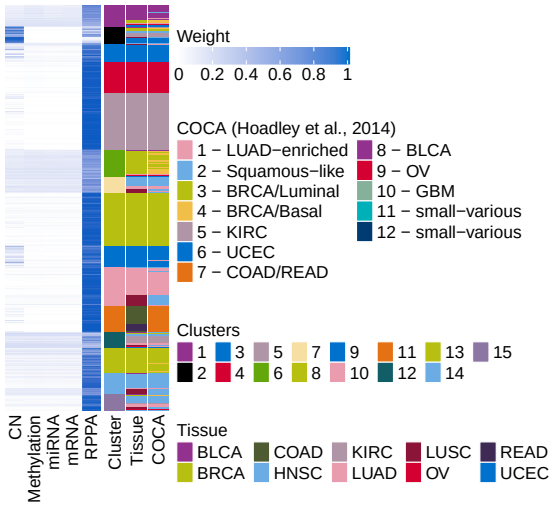


FIGURE C.45: Weights assigned by the multiple kernel *k*-means algorithm to each observations in each layer, where "CN" stands for copy number and "RPPA" for reverse phase protein array. The weights assigned on average to the tumour samples in each layer are: copy number 8.1%, methylation 3.8%, miRNA 3.6%, mRNA 3.7%, protein 80.8%.

C.2.4 *Outcome-guided integration: additional figures and results*

First, we report some additional figures for the outcome-guided integration of Section 4.4, then we repeat the analysis with the reduced datasets obtained as described in Section C.2.1.

Comparison with the clusters identified by Hoadley et al. (2014)

In Figure C.46 are shown the correspondences between the clusters found in the main paper in the outcome-guided case and the clusters identified by Hoadley et al. (2014) using Cluster-Of-Clusters Analysis (COCA).

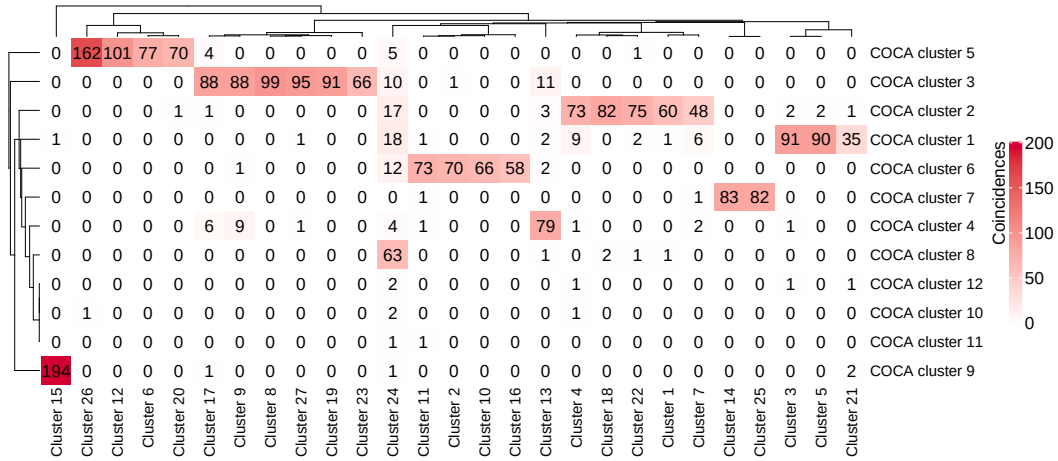


FIGURE C.46: Comparison between the clusters found combining the PSMs of each layer using the outcome-guided approach and those identified by Hoadley et al. (2014) using COCA.

Clustering structure in the data

Figures C.47, C.48, C.49, and C.50 show the four data layers where the rows have been sorted by final cluster.

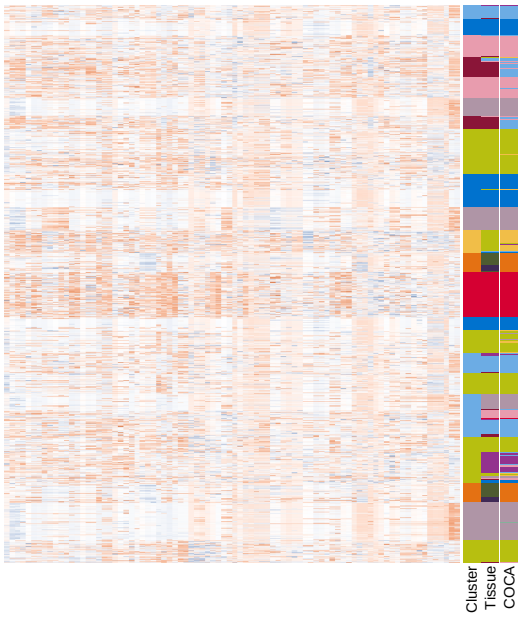


FIGURE C.47: Left: copy number data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

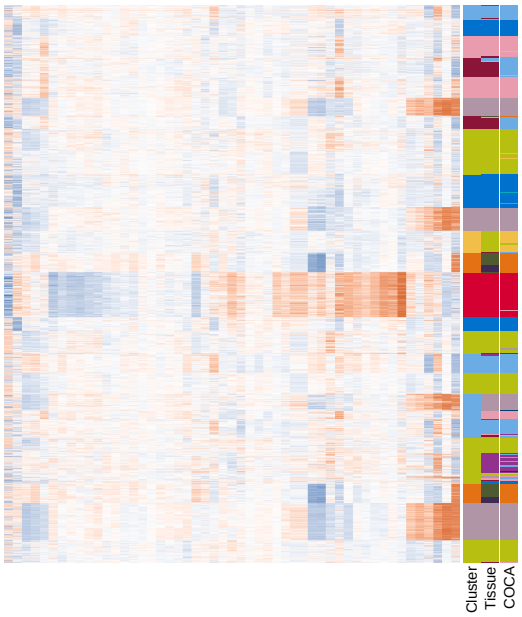


FIGURE C.48: Left: miRNA expression data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

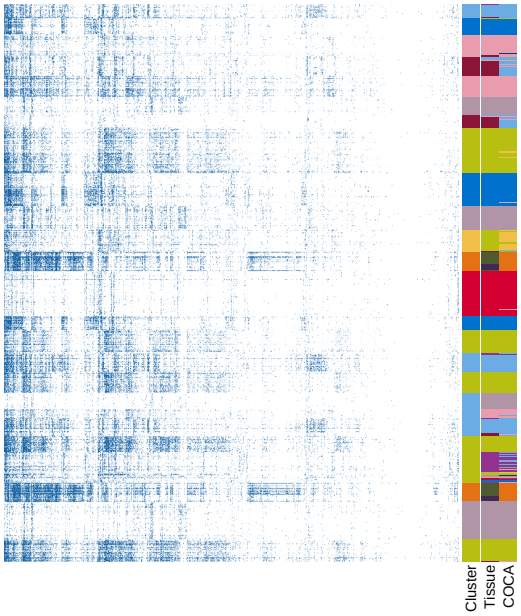


FIGURE C.49: Left: methylation data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

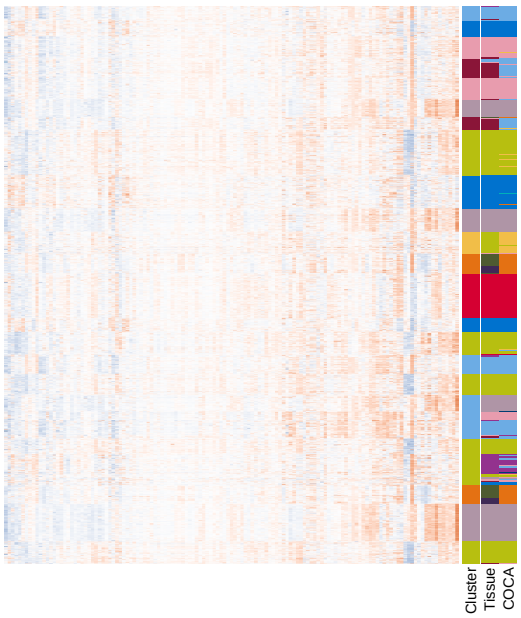
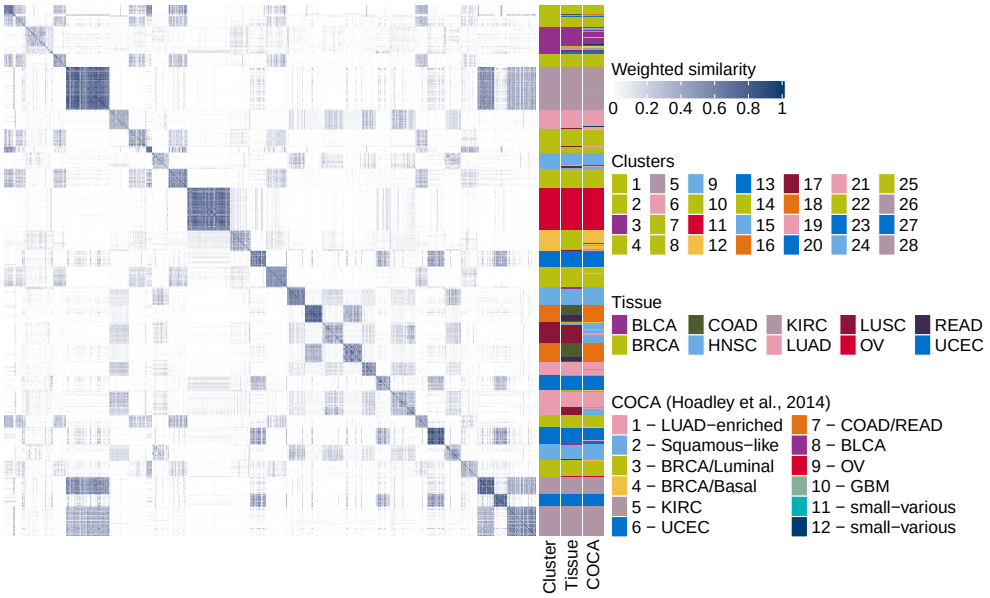


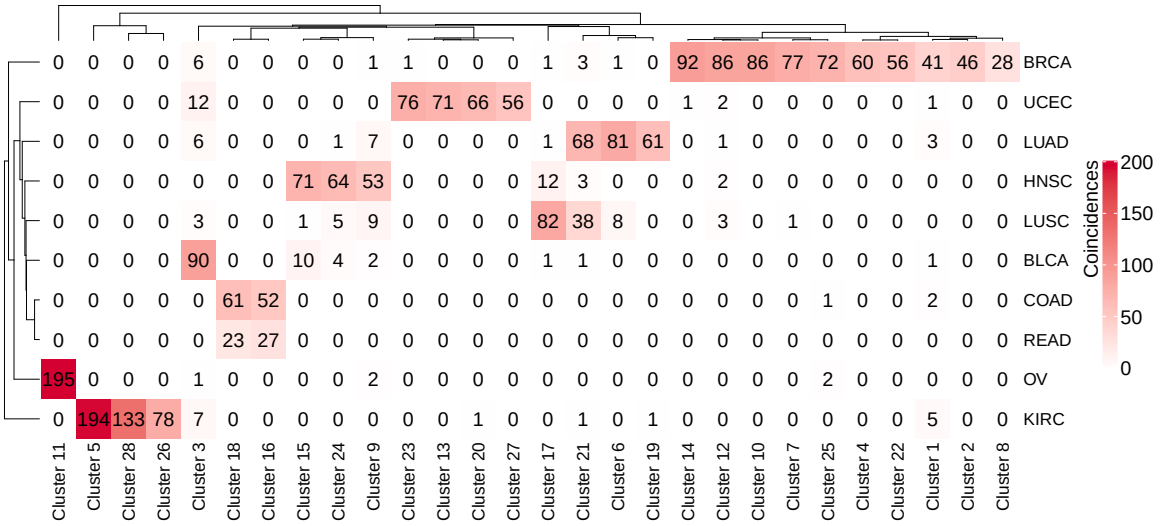
FIGURE C.50: Left: protein expression data (each row is a tumour sample). Right: final clusters, tissues of origin of each sample, and COCA clusters.

Outcome-guided integration after variable selection, $\alpha = 0.1$

The weights assigned on average to the tumour samples in each layer are: copy number 33.1%, methylation 16.3%, miRNA 34.1%, and protein 16.5%.



(A) Clusters and weighted kernel.



(B) Coincidence matrix.

FIGURE C.51: Outcome-guided multiplatform analysis of ten cancer types. (A) Weighted kernel, final clusters, tissues of origin, and COCA clusters. (B) Coincidence matrix comparing the tissue of origin of the tumour samples with the clusters.

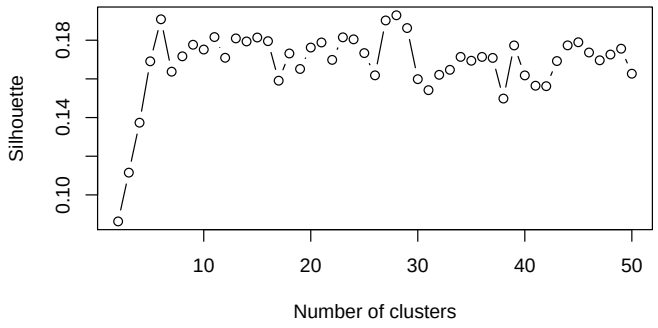


FIGURE C.52: Average silhouette for number of clusters going from 2 to 50.

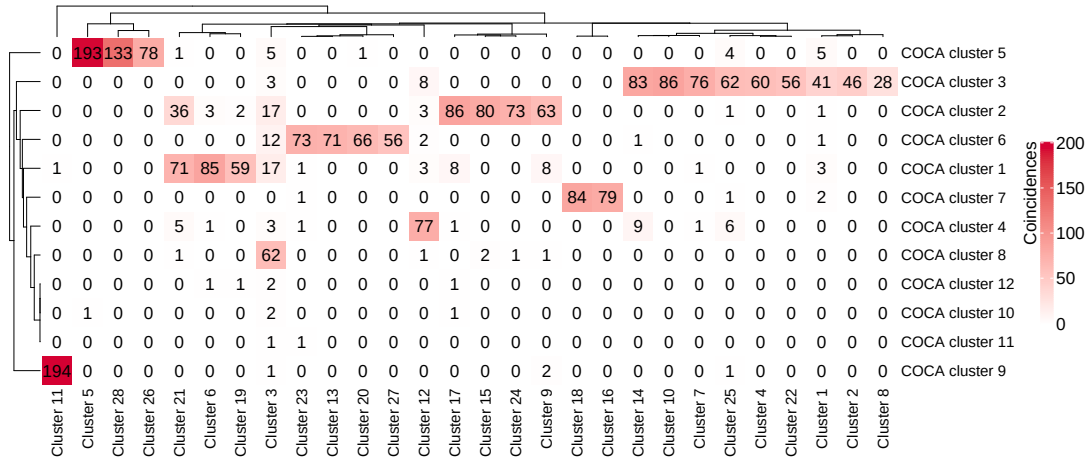
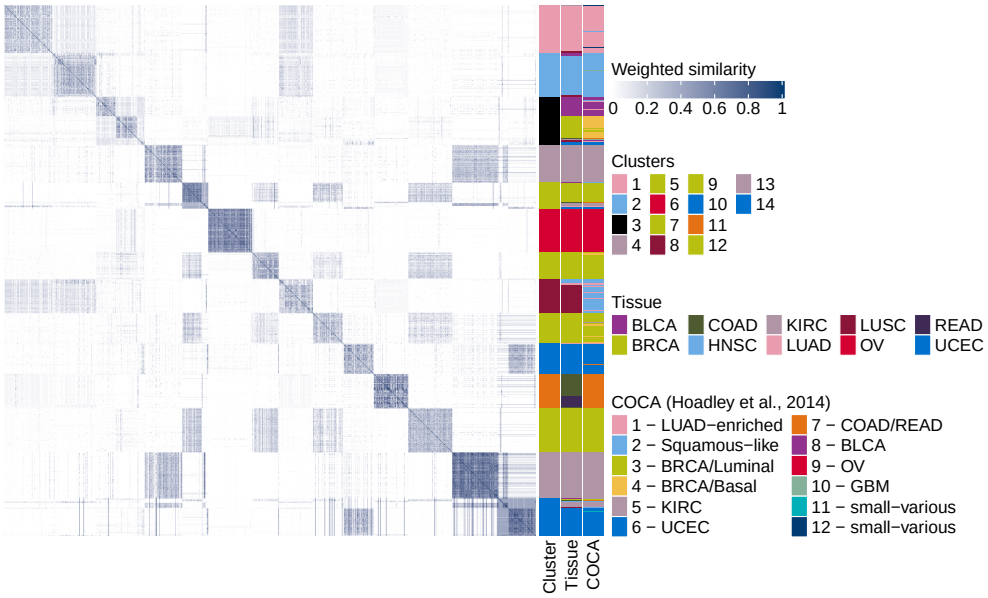


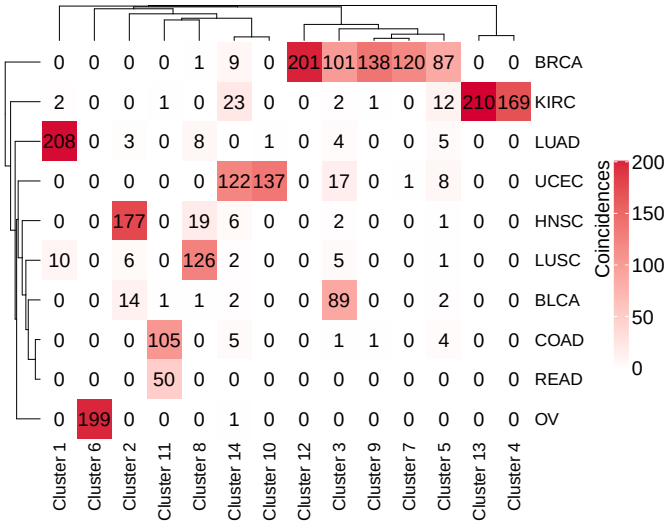
FIGURE C.53: Comparison between the clusters found combining the PSMs of each layer using the outcome-guided approach and those identified by Hoadley et al. (2014) using COCA.

Outcome-guided integration after variable selection, $\alpha = 0.5$

The weights assigned on average to the tumour samples in each layer are: copy number 34.2%, mRNA 17.7%, methylation 7.2%, miRNA 27.4%, protein 13.5.



(A) Clusters and weighted kernel.



(B) Coincidence matrix.

FIGURE C.54: Outcome-guided multiplatform analysis of ten cancer types. (A) Weighted kernel, final clusters, tissues of origin, and COCA clusters. (B) Coincidence matrix comparing the tissue of origin of the tumour samples with the clusters.

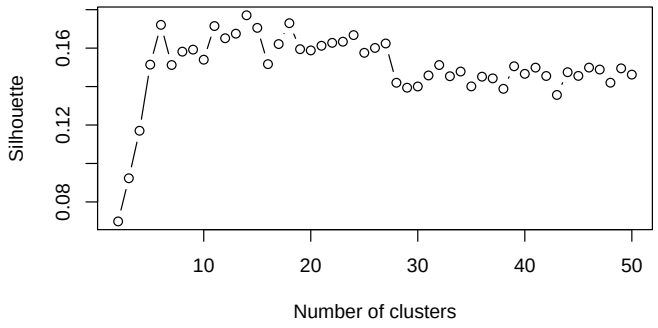


FIGURE C.55: Average silhouette for number of clusters going from 2 to 50.

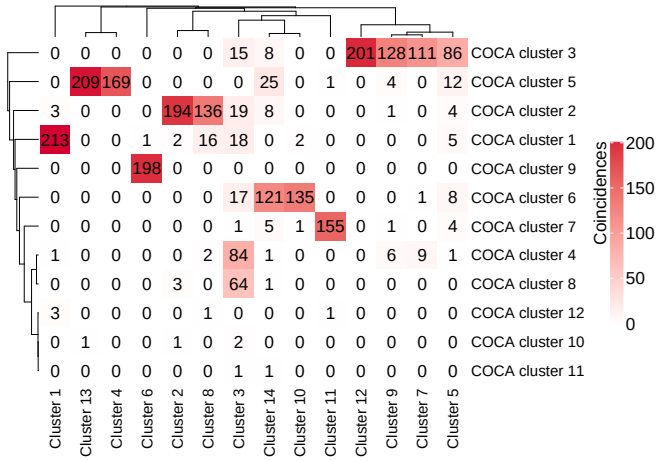
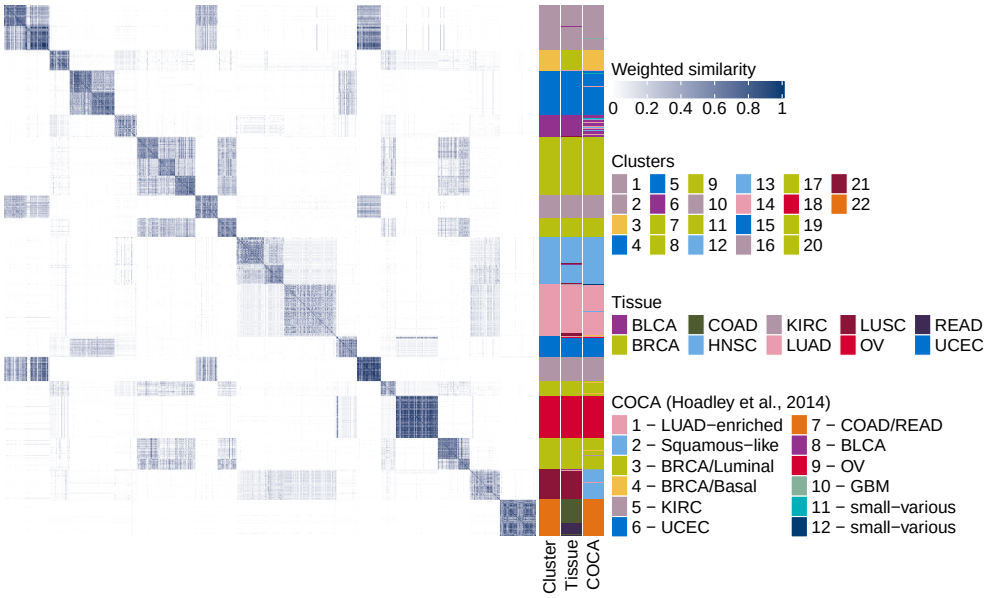


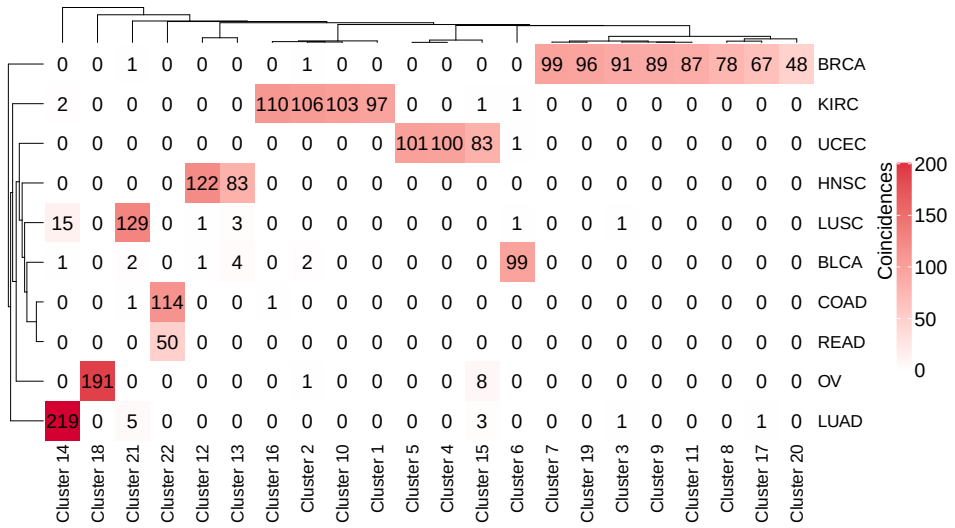
FIGURE C.56: Comparison between the clusters found combining the PSMs of each layer using the outcome-guided approach and those identified by Hoadley et al. (2014) using COCA.

Outcome-guided integration after variable selection, $\alpha = 1$

The weights assigned on average to the tumour samples in each layer are: copy number 0%, mRNA 37.6%, methylation 25.2%, miRNA 24.5%, protein 12.7%.



(A) Clusters and weighted kernel.



(B) Coincidence matrix.

FIGURE C.57: Outcome-guided multiplatform analysis of ten cancer types. (A) Weighted kernel, final clusters, tissues of origin, and COCA clusters. (B) Coincidence matrix comparing the tissue of origin of the tumour samples with the clusters.

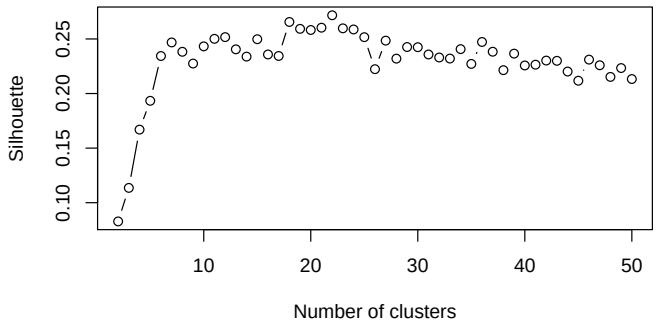


FIGURE C.58: Average silhouette for number of clusters going from 2 to 50.

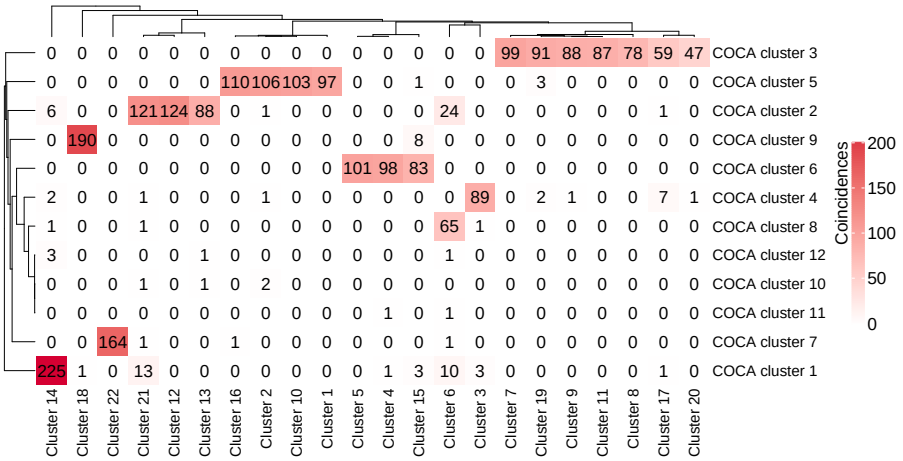
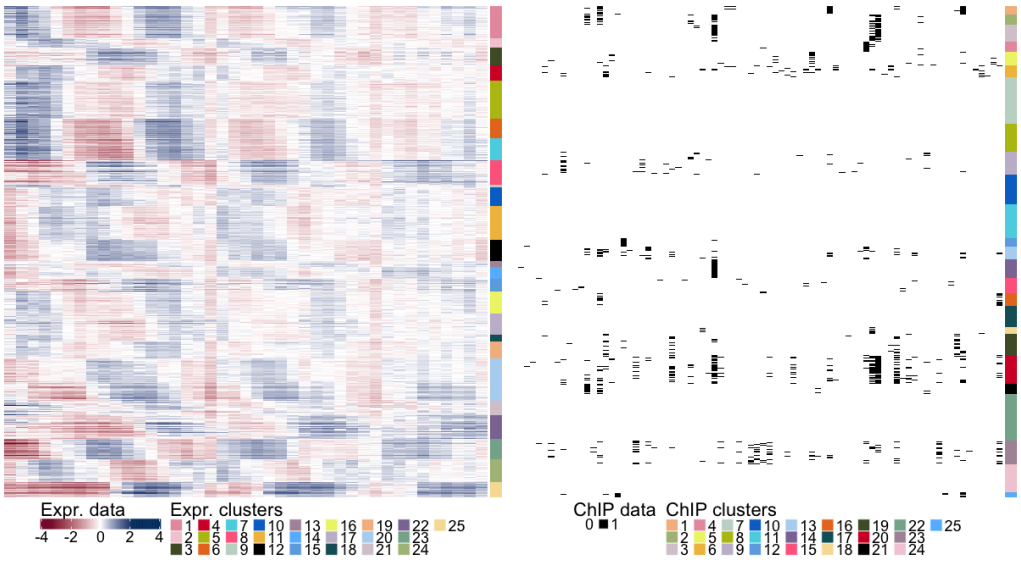


FIGURE C.59: Comparison between the clusters found combining the PSMs of each layer using the outcome-guided approach and those identified by Hoadley et al. (2014) using COCA.

C.3 TRANSCRIPTIONAL MODULE DISCOVERY

We present here some additional figures for the transcriptional module discovery application of Section 4.5.

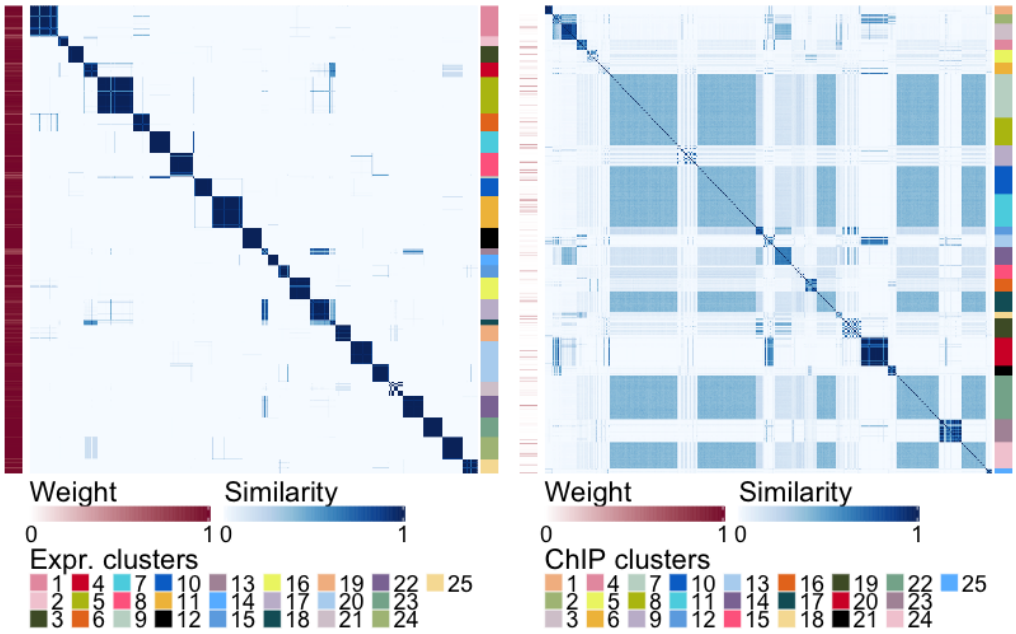
Figures C.60 and C.61 show the initial data, the clusterings obtained on each dataset individually and the PSMs. The cophenetic correlation coefficients are 0.953685 for the expression data and 0.9841434 for the ChIP data. 3.5% of the weight is assigned on average to the ChIP data, and the remaining 96.5% to the expression data.



(A) Expression data. (B) ChIP-chip data.

FIGURE C.60: Clusters obtained on each dataset separately. The ordering of the rows is different in the two figures. (A) Left: expression data. Each row corresponds to a gene and each column to a different time point. Right: clusters obtained using the expression data only. (B) Left: ChIP-chip data. Each row corresponds to a gene and each column to a transcriptional regulator. Right: clusters obtained using the ChIP-chip data only.

In Figure C.62 are reported the values of the average silhouette for different values of the number of clusters K. We choose $K = 25$, which gives the highest value of the silhouette.



(A) Expression data. (B) ChIP-chip data.

FIGURE C.61: PSMs and clusterings obtained via kernel k-means on each dataset separately. The ordering of the observations is different in the two figures. (A) Left: weights assigned to the expression data by KLIC. Centre: PSM of the expression data. Right: clusters obtained using the expression data only. (B) Right: weights assigned to the ChIP-chip data by KLIC. Centre: PSM of the ChIP-chip data. Right: clusters obtained using the ChIP-chip data only.

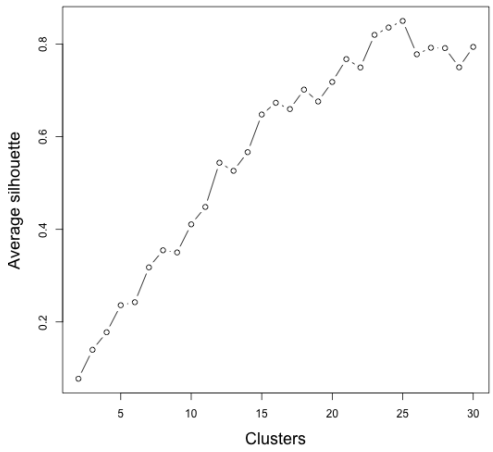


FIGURE C.62: Plot of silhouette for different numbers of clusters for the integration of the datasets of Harbison *et al.* and Granovskaia *et al.*. The maximum value is attained at $K = 25$.

D

APPENDIX TO CHAPTER 5

This appendix contains additional figures and results for the cluster analysis of the CMS data presented in Chapter 5.

D.1 APPLYING KLIC TO THE CARDIOMETABOLIC DISEASE DATA

Sections D.1.1 and D.1.2 contain some figures and tables summarising the output of the MCMC algorithms run on the full and reduced CMS data layers respectively. Sections D.1.3 and D.1.4 contain additional figures for the unsupervised and outcome-guided analyses of the CMS data respectively.

D.1.1 *MCMC convergence assessment: full dataset*

As for the pan-cancer data analysis (Appendix C), we run five MCMC chains for 50,000 iterations, with a burn-in period of 25,000 iterations and thinning of 5. For each set of five chains, we check the Vats-Knudson $\hat{R}$ (Vats and Knudson, 2018) with parameters $\epsilon = 0.1$ and $\alpha = 0.1$ to assess the convergence of the mass parameter. The PSMs obtained for the five chains are summarised into one by taking the average.

MCMC convergence assessment

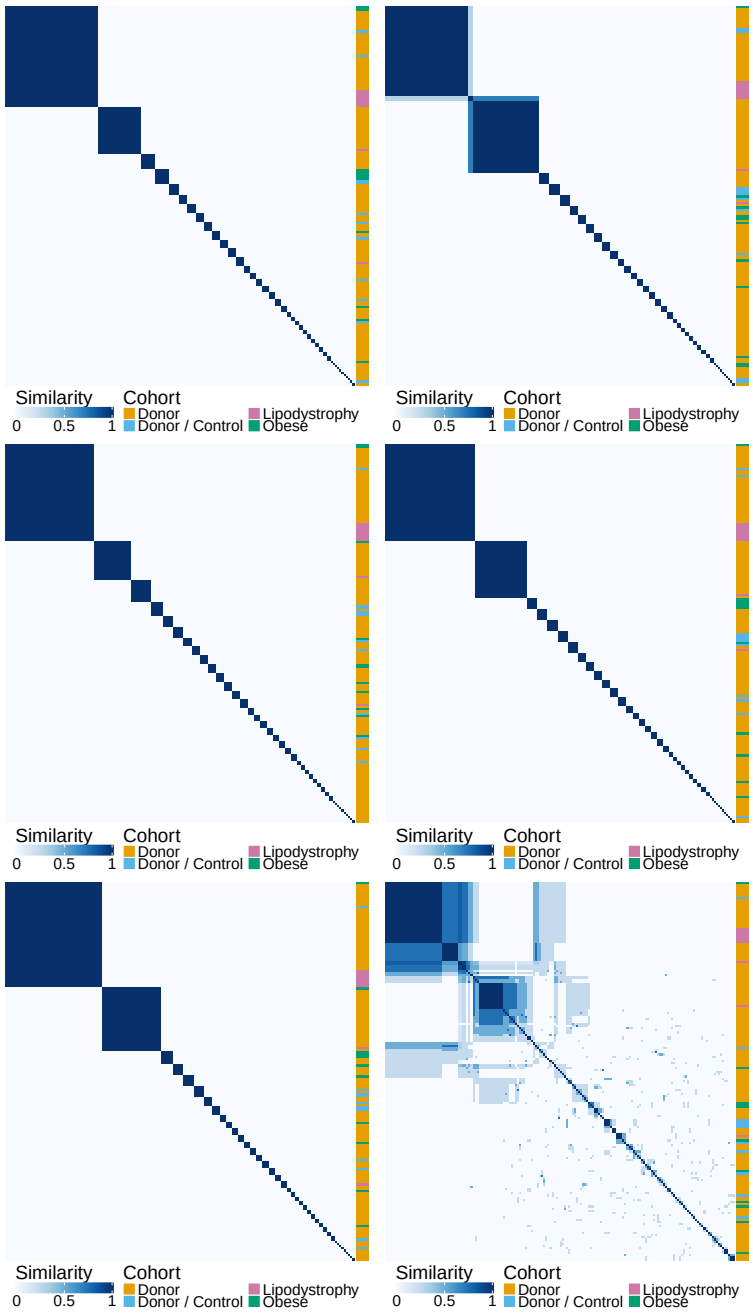


FIGURE D.1: Five PSMs of the ChIP-seq data (monocytes) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.22	0.31	0.27	0.26
Chain 2	1	0.24	0.56	0.35
Chain 3		1	0.30	0.26
Chain 4			1	0.33

TABLE D.1: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

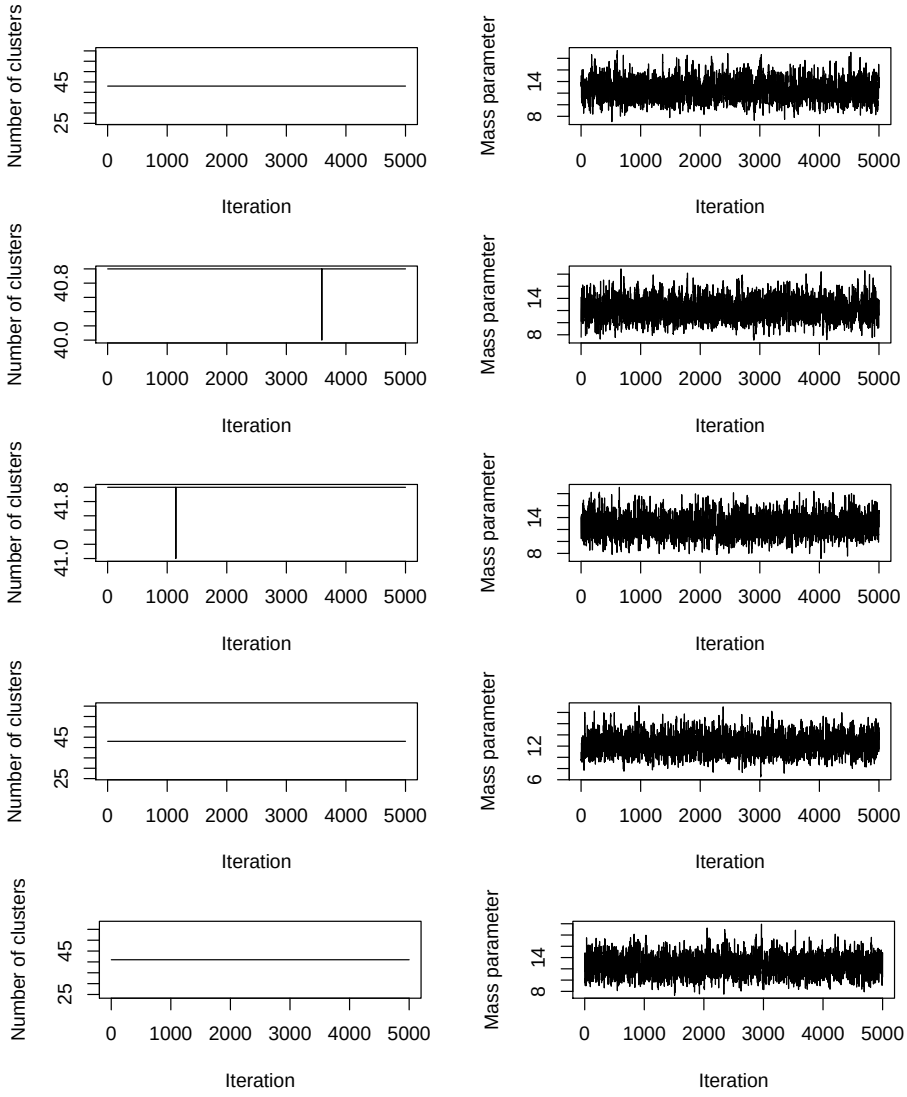


FIGURE D.2: MCMC convergence assessment, ChIP-seq data (monocytes).

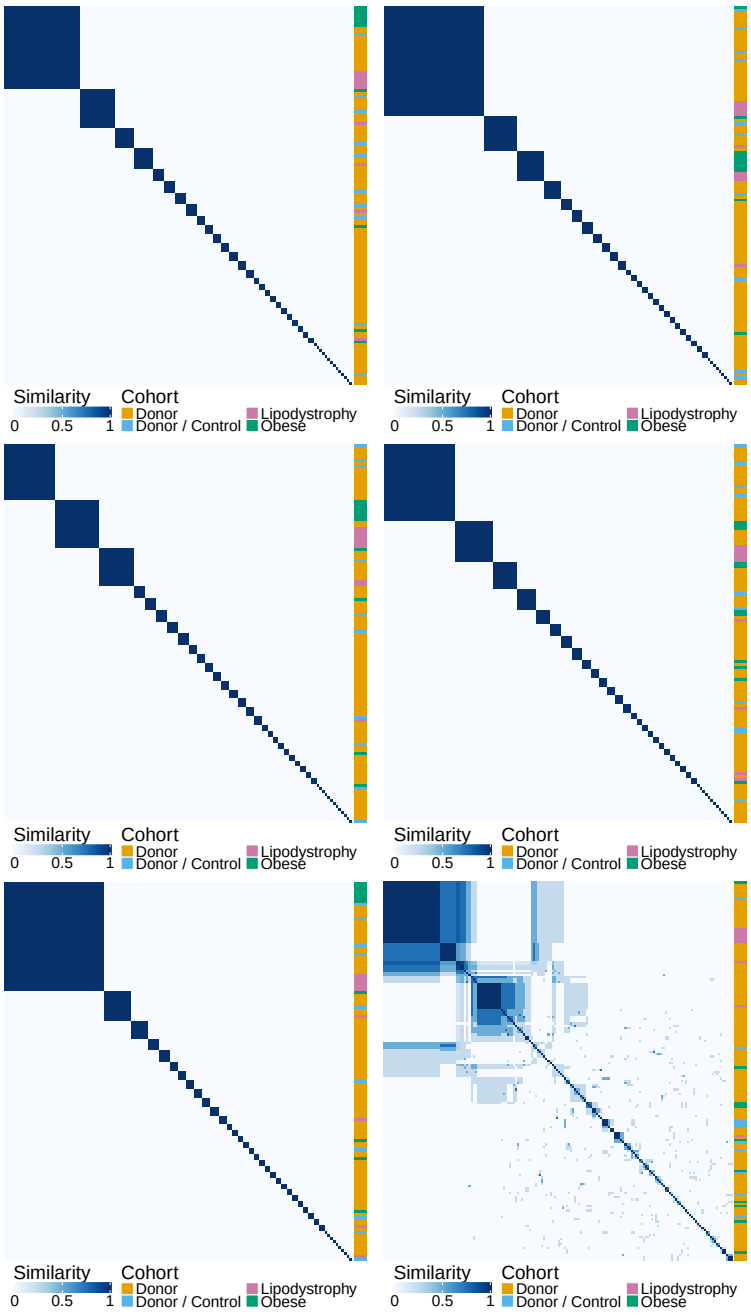


FIGURE D.3: Five PSMs of the ChIP-seq data (neutrophils) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.20	0.20	0.10	0.80
Chain 2	1	0.18	0.16	0.18
Chain 3		1	0.12	0.12
Chain 4			1	0.17

TABLE D.2: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

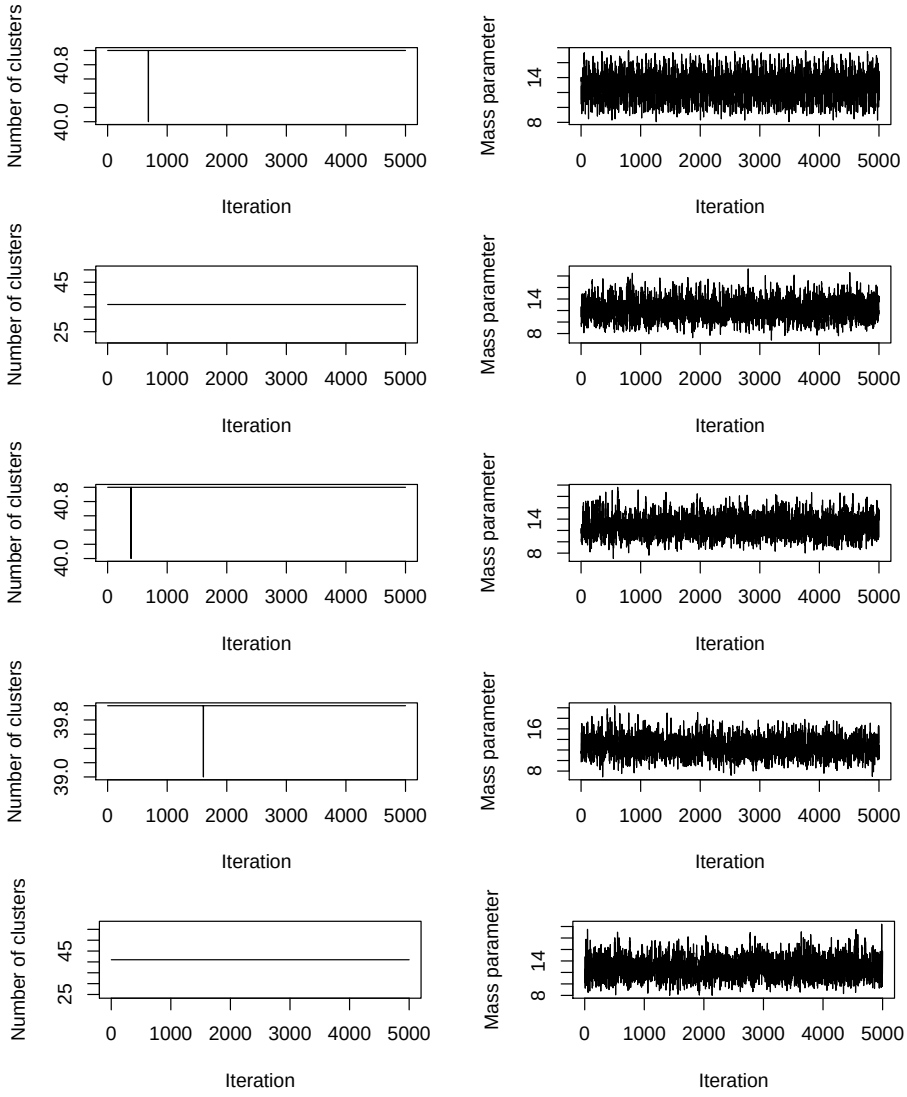


FIGURE D.4: MCMC convergence assessment, ChIP-seq data (neutrophils).

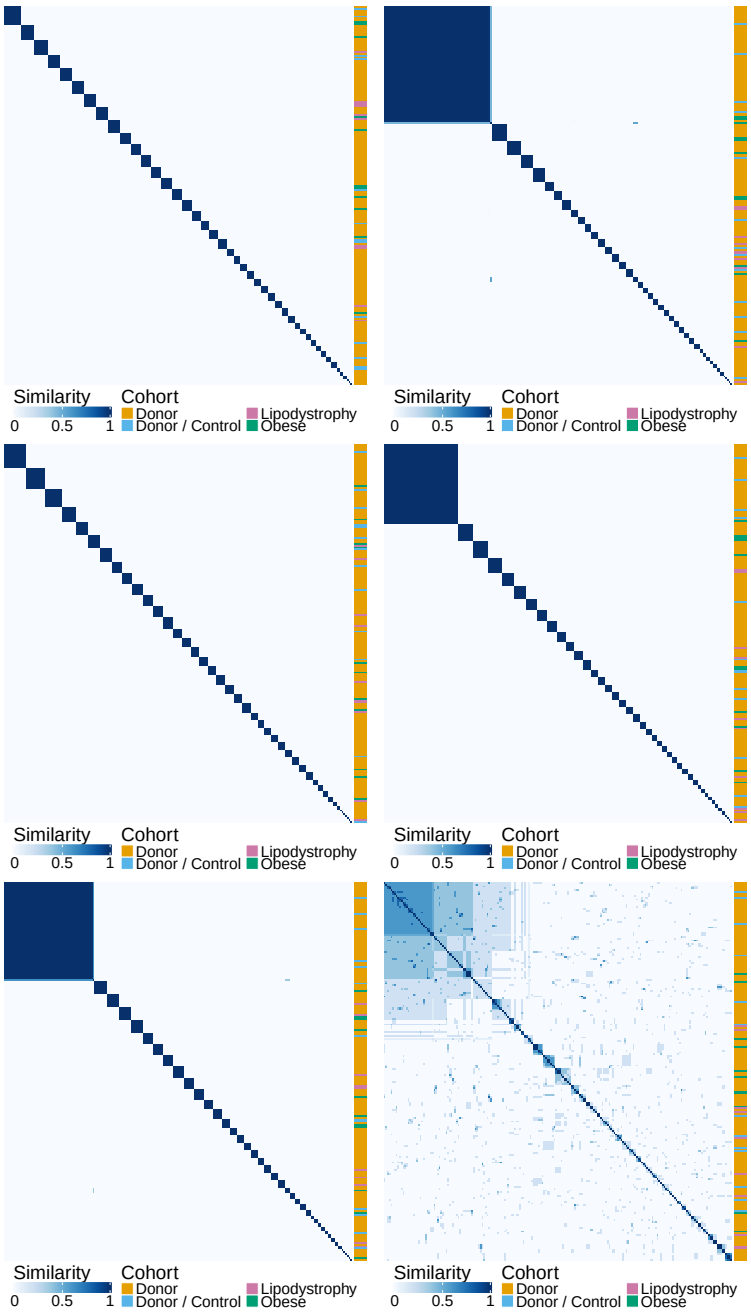


FIGURE D.5: Five PSMs of the RNA-seq (monocytes) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	**Chain 3**	**Chain 4**	**Chain 5**
Chain 1	0.08	0.20	0.14	0.07
Chain 2	1	0.04	0.21	0.10
Chain 3		1	0.09	0.07
Chain 4			1	0.07

TABLE D.3: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

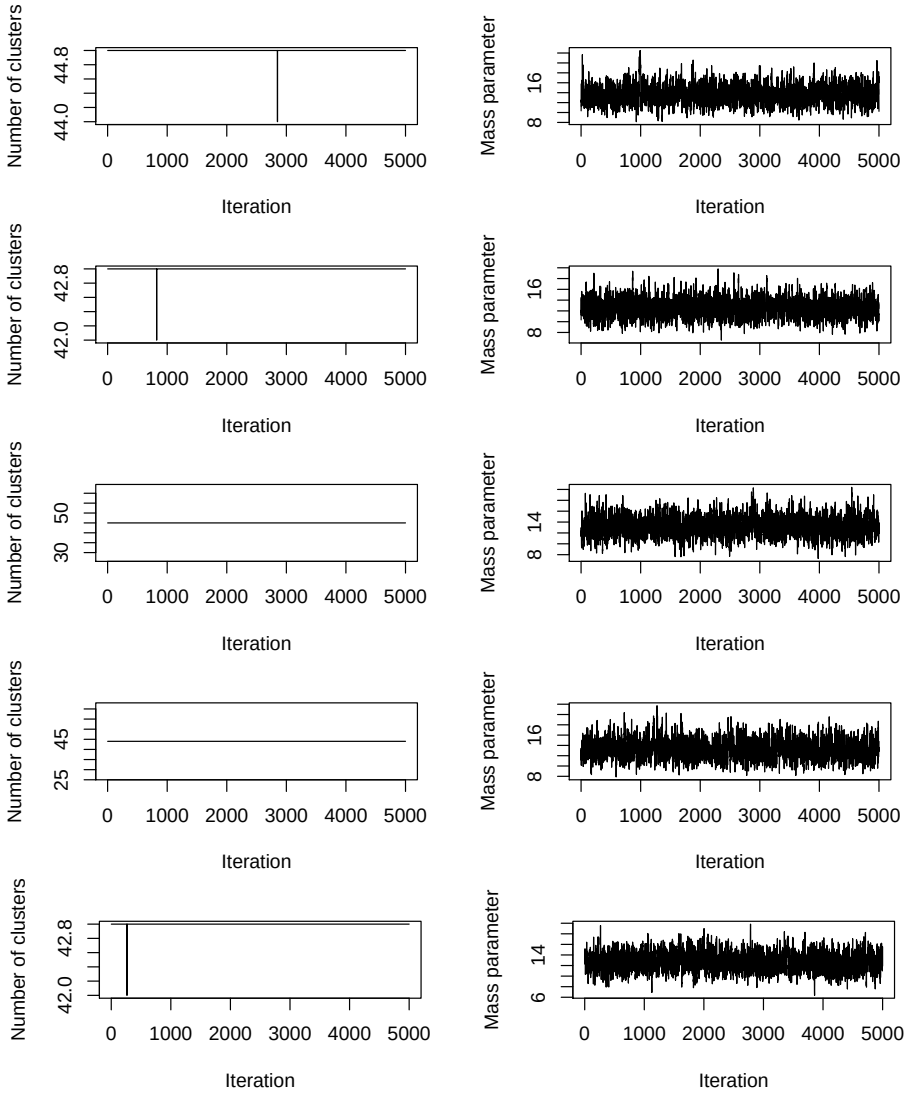


FIGURE D.6: MCMC convergence assessment, RNA-seq data (monocytes).

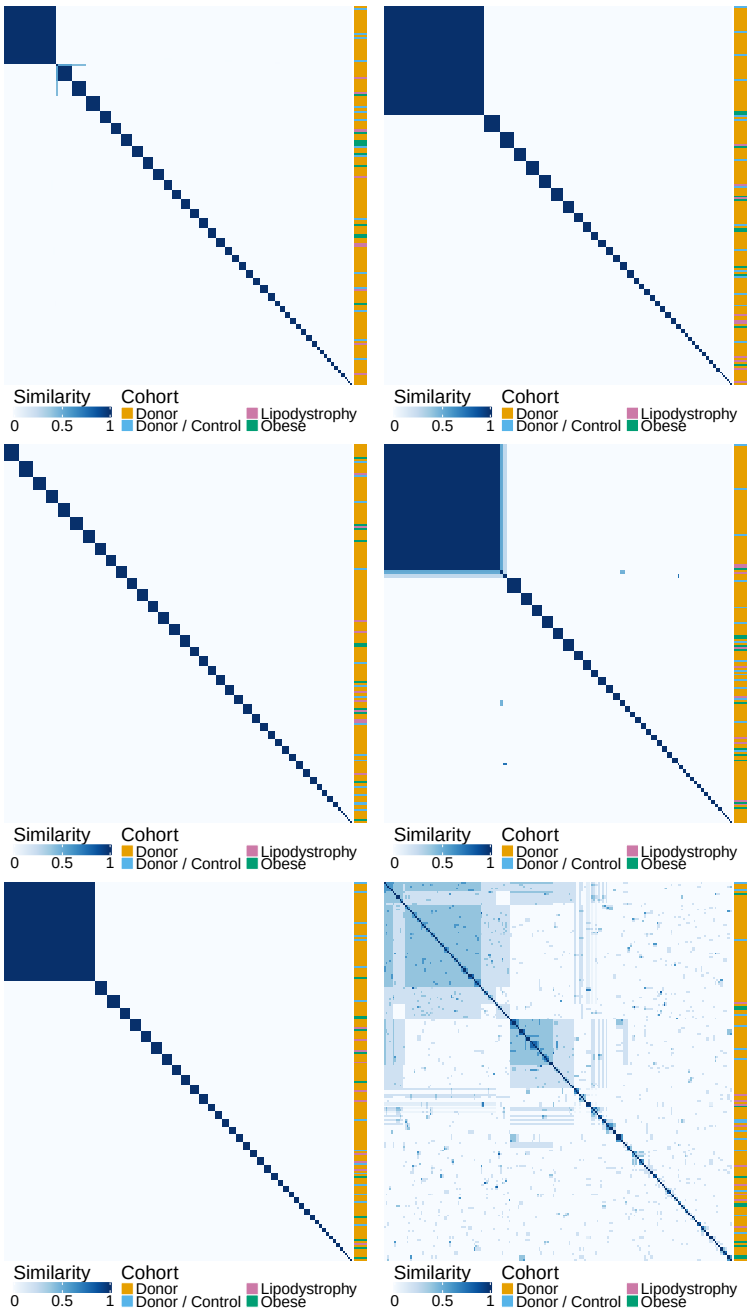


FIGURE D.7: Five PSMs of the RNA-seq (neutrophils) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.11	0.13	0.04	0.16
Chain 2	1	0.06	0.25	0.03
Chain 3		1	0.05	0.06
Chain 4			1	0.04

TABLE D.4: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

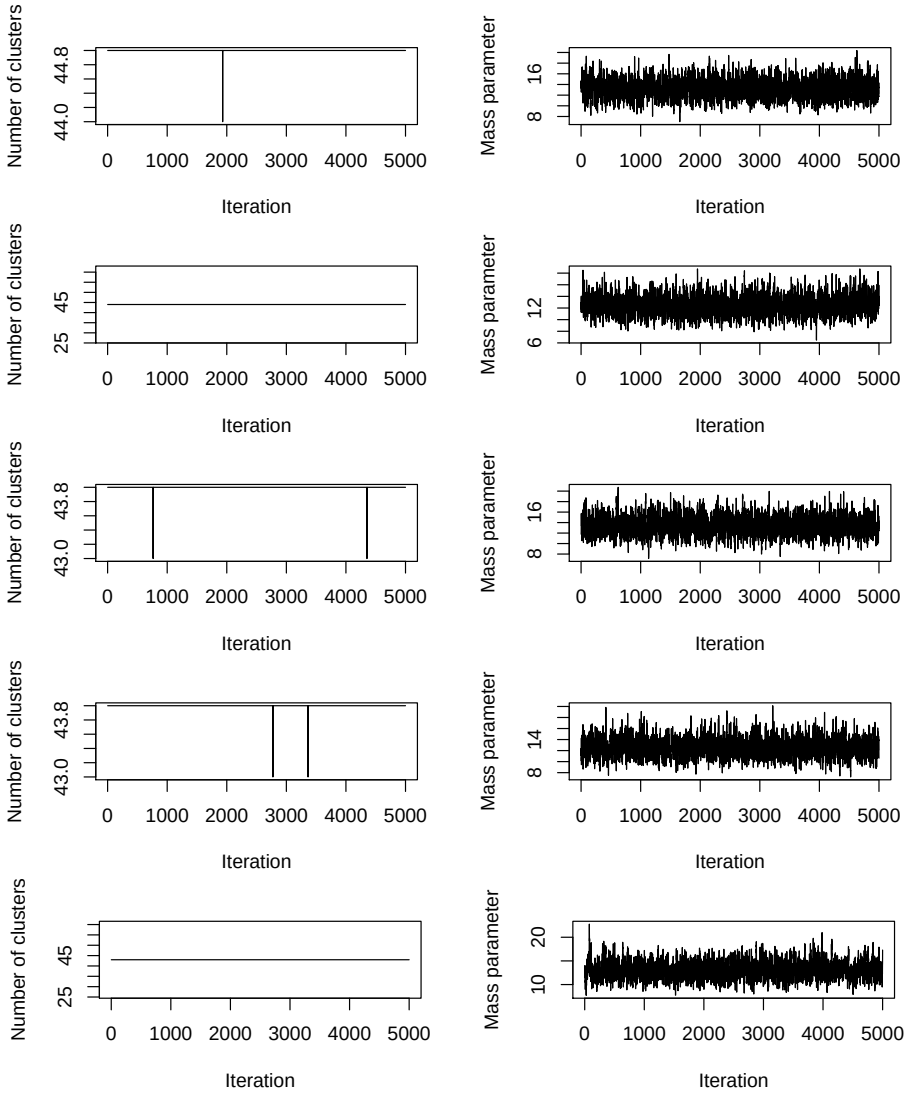


FIGURE D.8: MCMC convergence assessment, RNA-seq data (neutrophils).

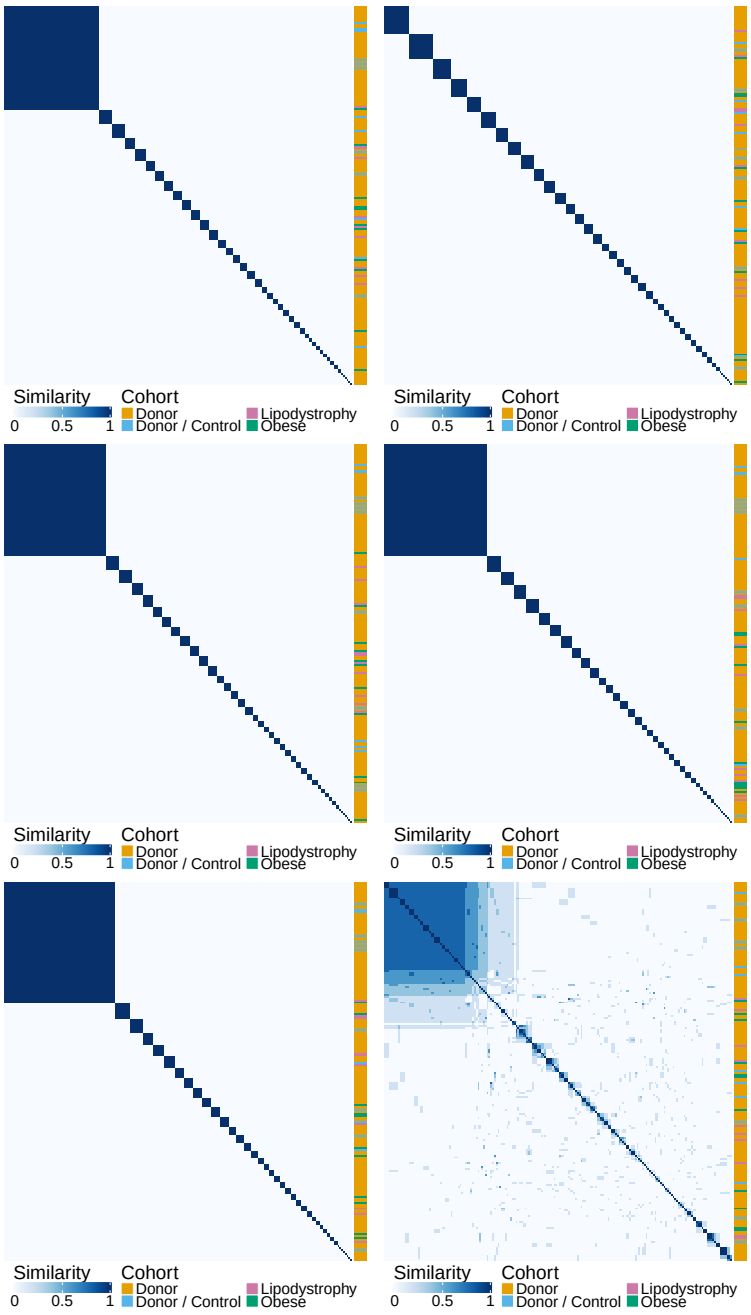


FIGURE D.9: Five PSMs of the methylation (monocytes) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.13	0.23	0.25	0.18
Chain 2	1	0.10	0.20	0.06
Chain 3		1	0.26	0.18
Chain 4			1	0.20

TABLE D.5: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

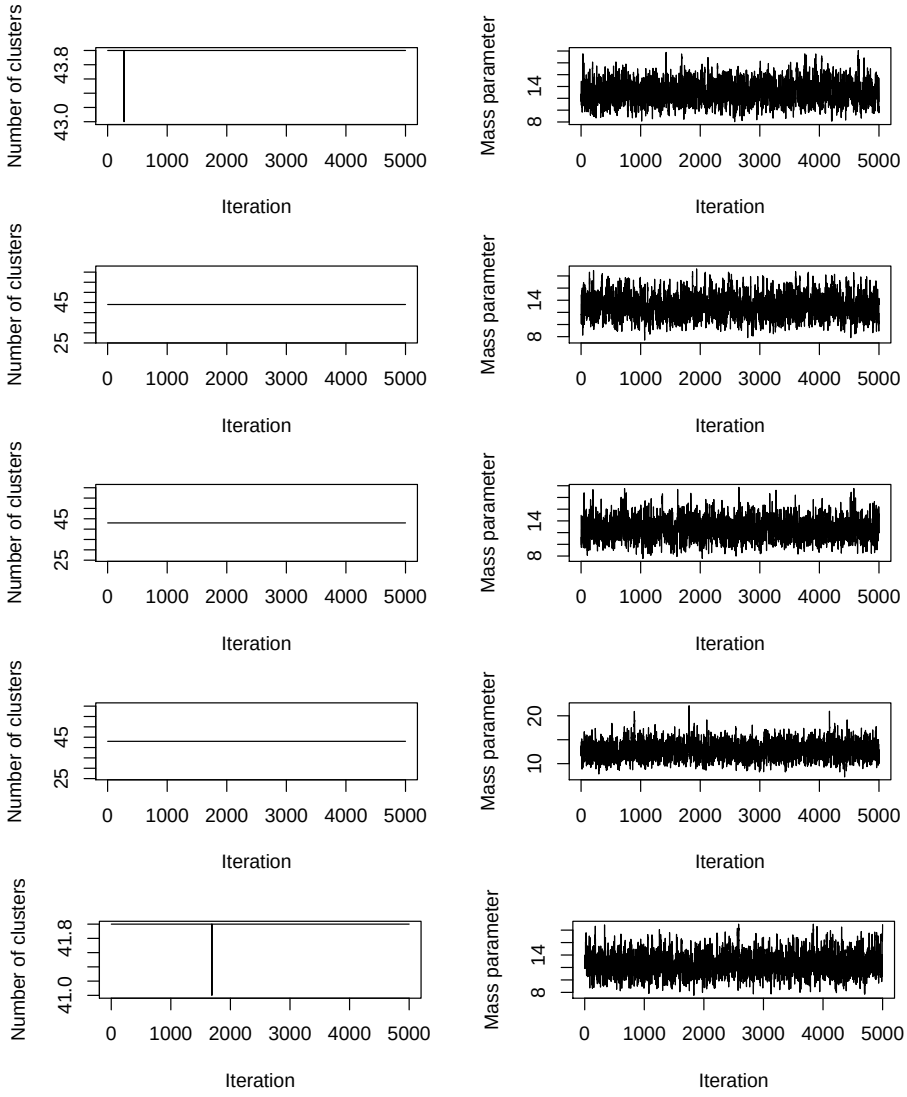


FIGURE D.10: MCMC convergence assessment, methylation data (monocytes).

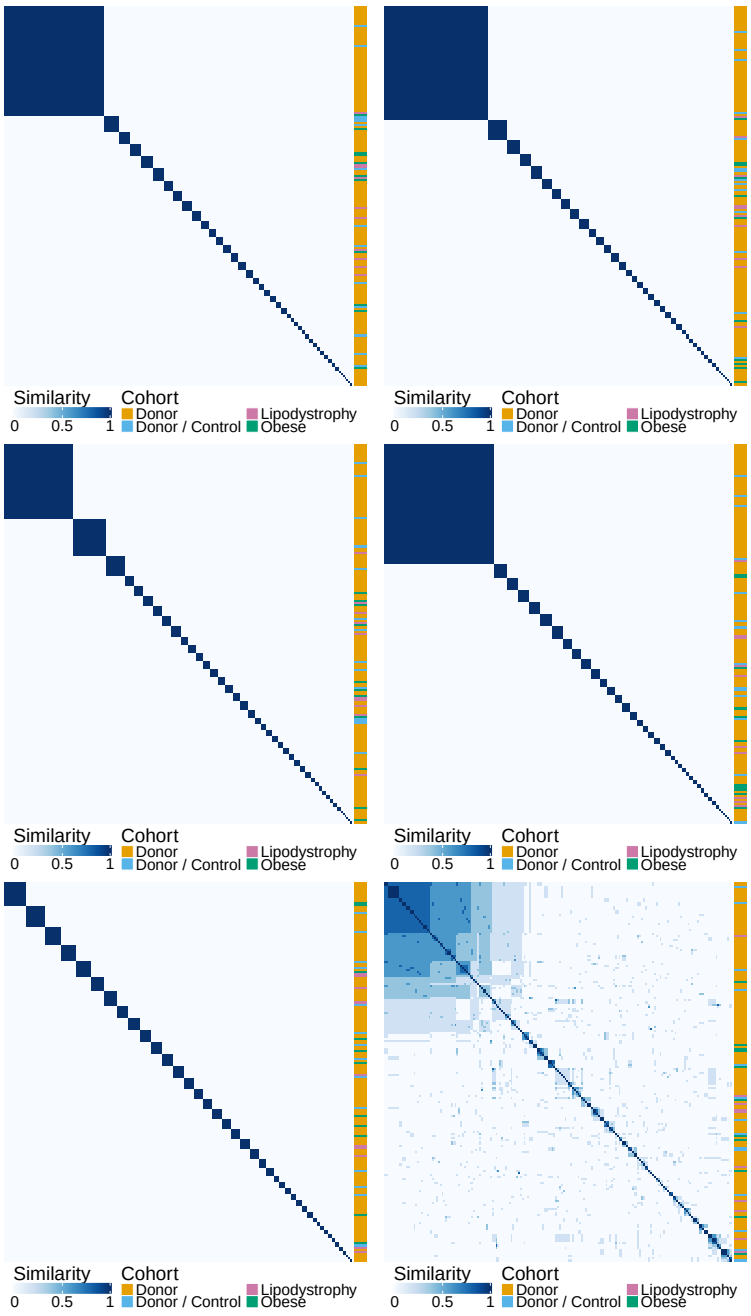


FIGURE D.11: Five PSMs of the methylation (neutrophils) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.17	0.11	0.12	0.09
Chain 2	1	0.17	0.33	0.06
Chain 3		1	0.14	0.08
Chain 4			1	0.05

TABLE D.6: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

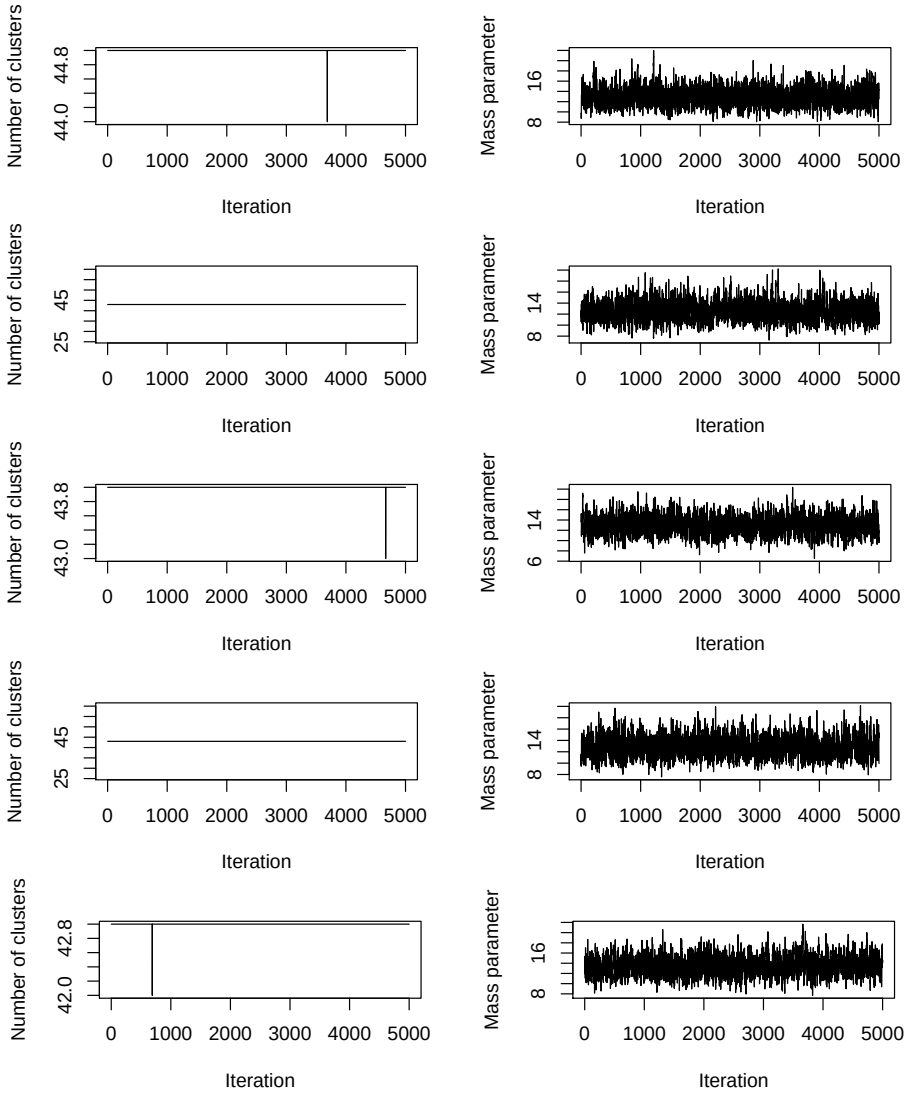


FIGURE D.12: MCMC convergence assessment, methylation data (neutrophils).

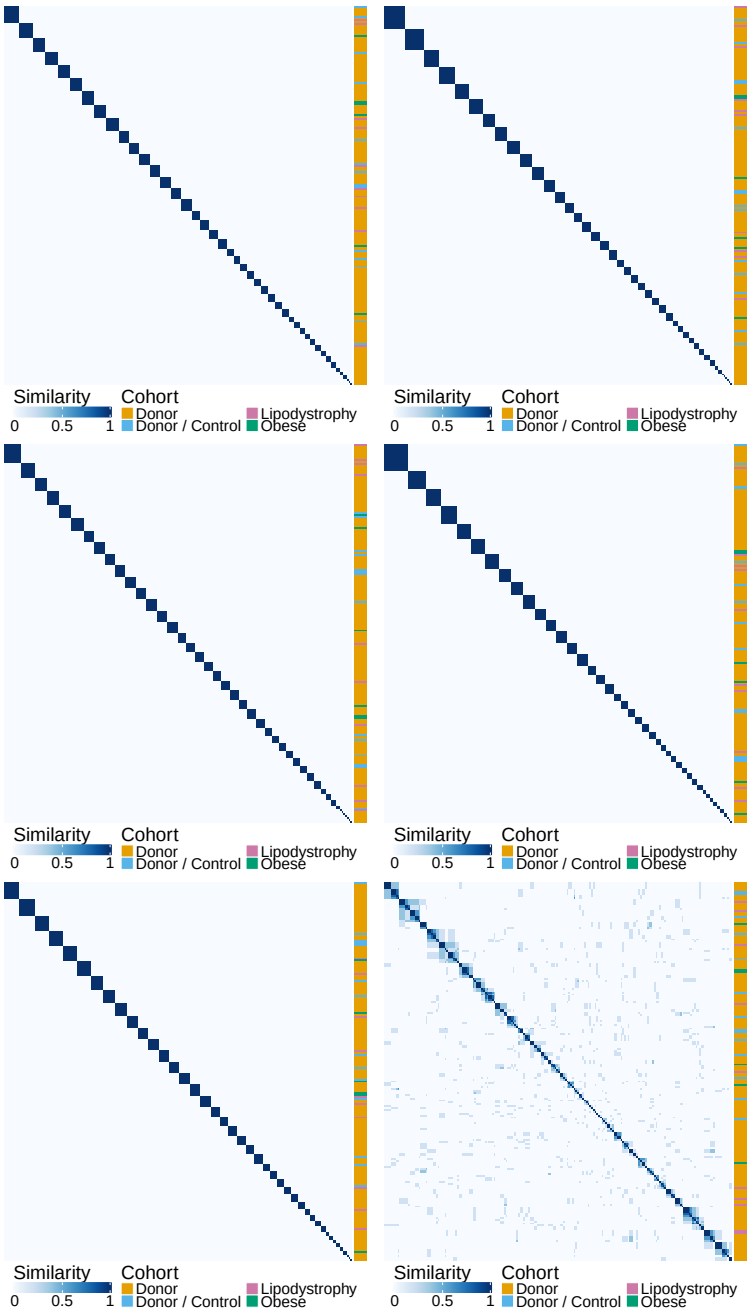


FIGURE D.13: Five PSMs of the metabolites and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.42	0.25	0.22	0.20
Chain 2	1	0.20	0.46	0.18
Chain 3		1	0.13	0.14
Chain 4			1	0.14

TABLE D.7: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

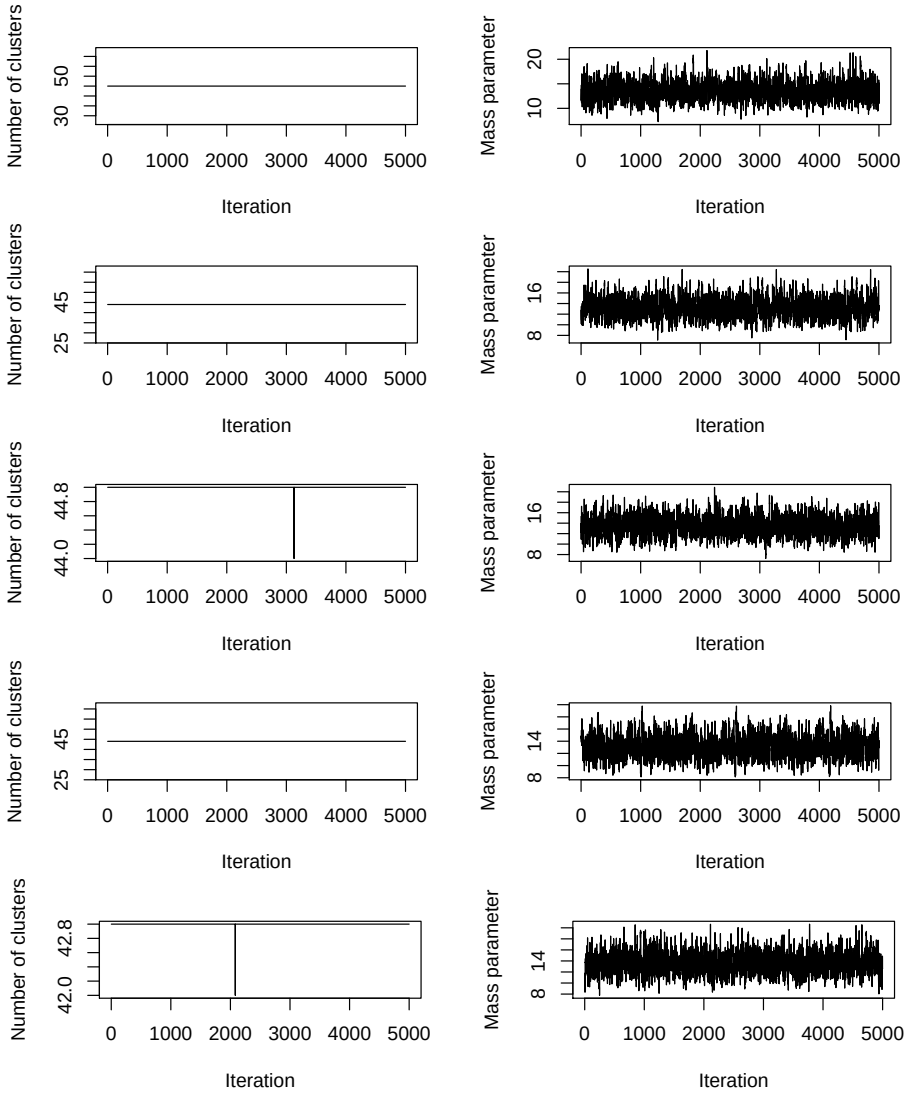


FIGURE D.14: MCMC convergence assessment, metabolite data.

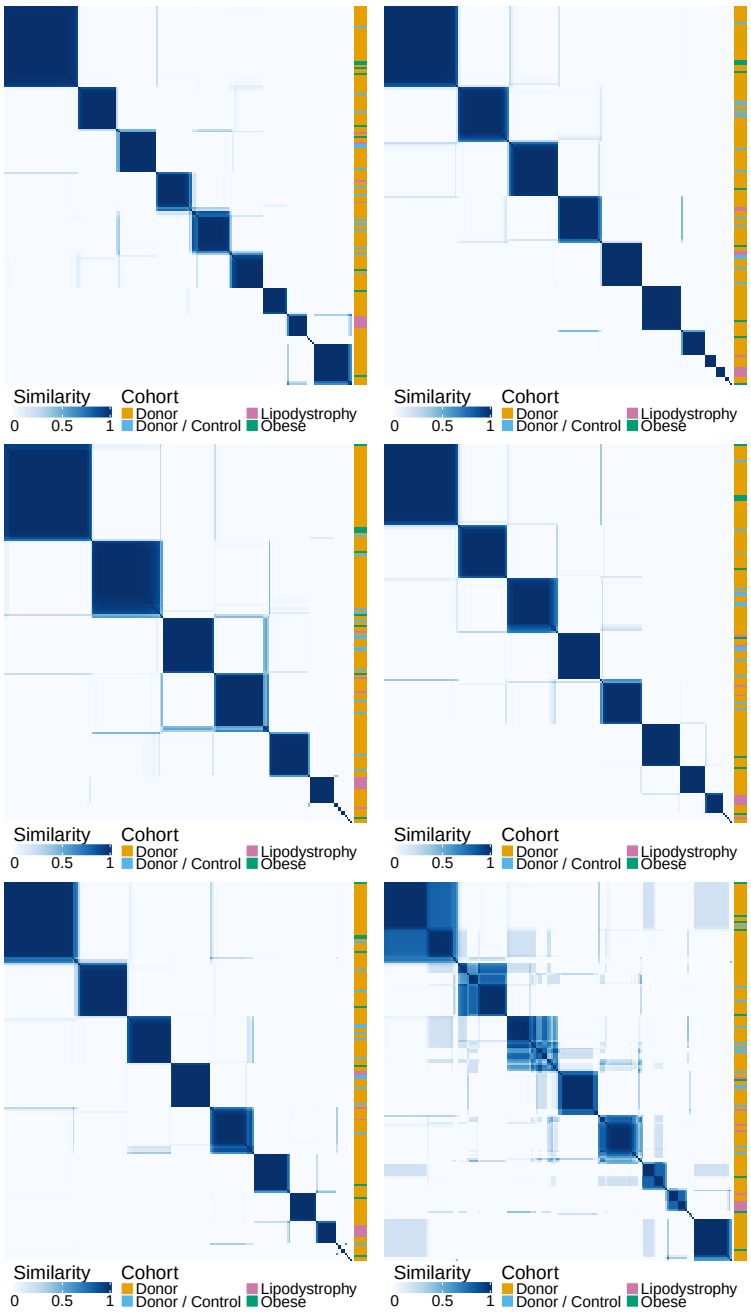


FIGURE D.15: Five PSMs of the lipids and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.82	0.38	0.83	0.80
Chain 2	1	0.42	0.95	0.90
Chain 3		1	0.42	0.40
Chain 4			1	0.93

TABLE D.8: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

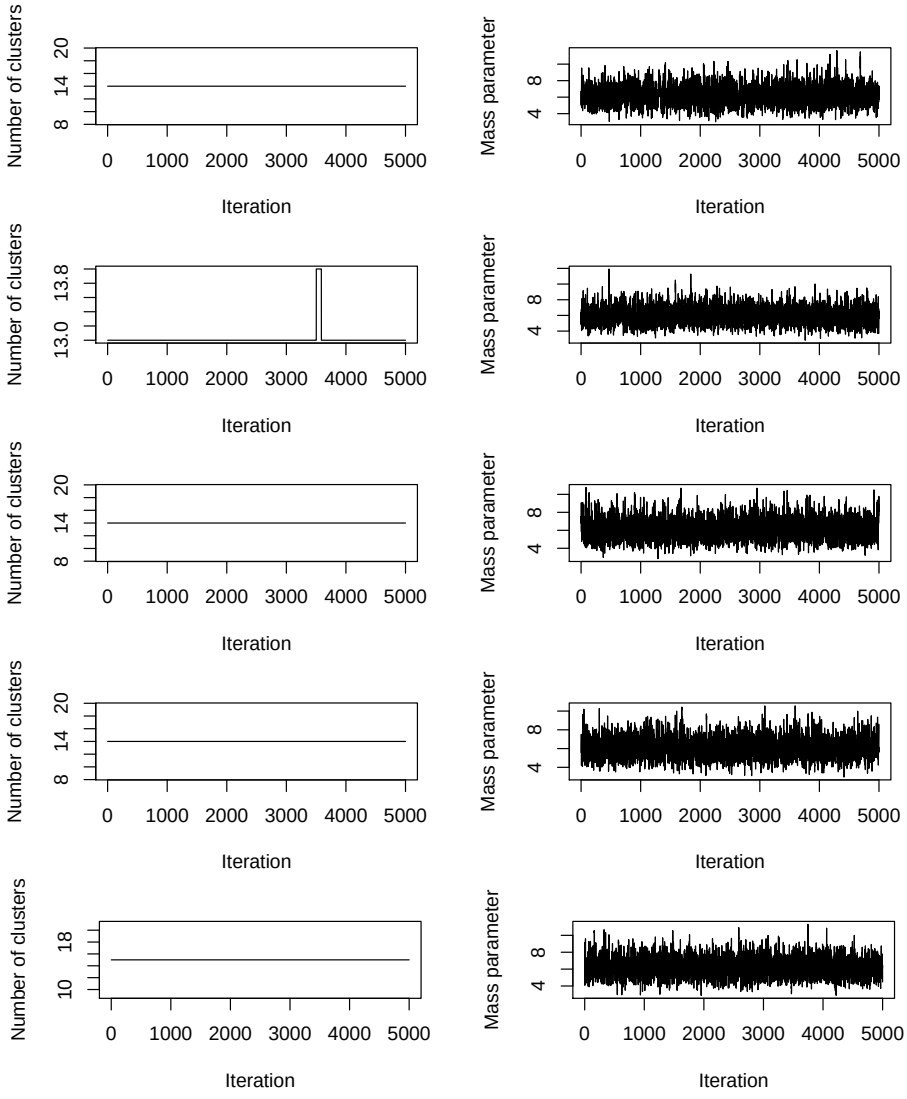


FIGURE D.16: MCMC convergence assessment, lipids data.

D.1.2 *MCMC convergence assessment: reduced dataset*

Again, we assess the convergence of the MCMC chains first, and then combine them to obtain one PSM for each 'omic layer.

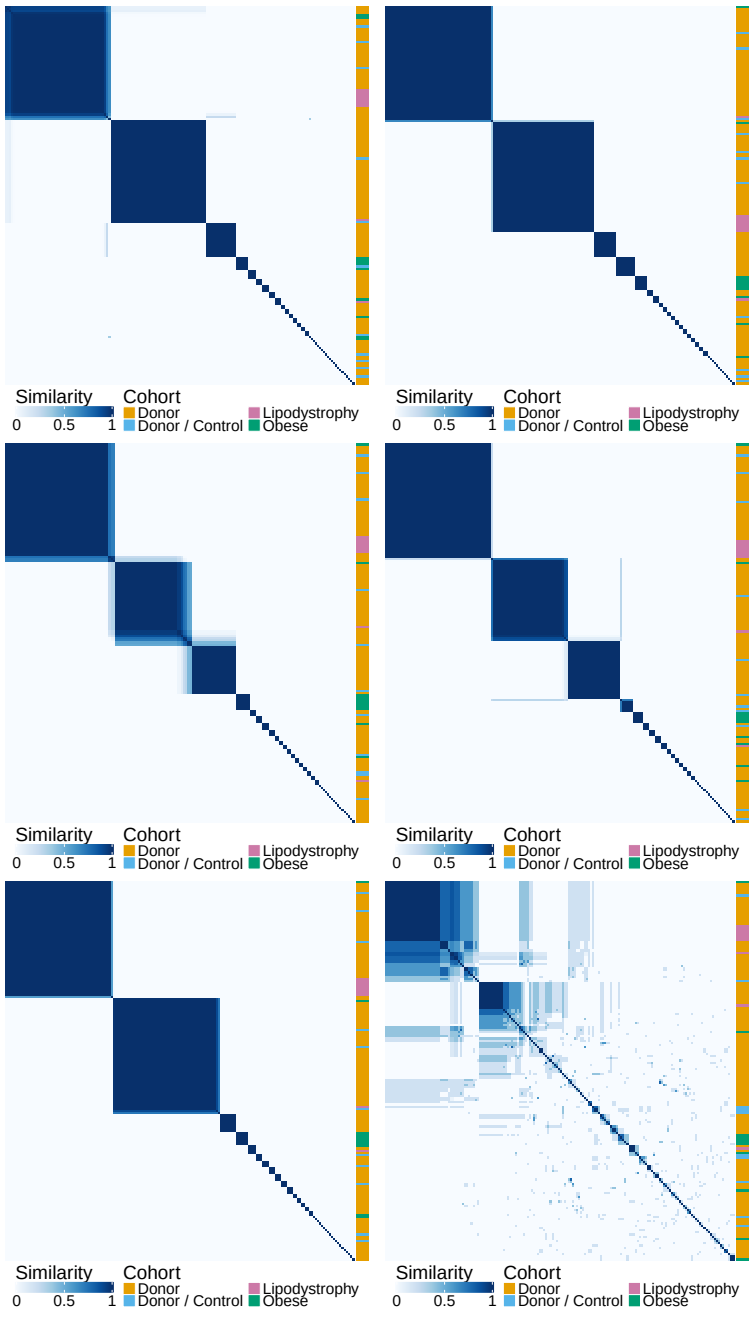


FIGURE D.17: Five PSMs of the ChIP-seq data (monocytes) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.33	0.27	0.26	0.51
Chain 2	1	0.24	0.21	0.32
Chain 3		1	0.45	0.34
Chain 4			1	0.33

TABLE D.9: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

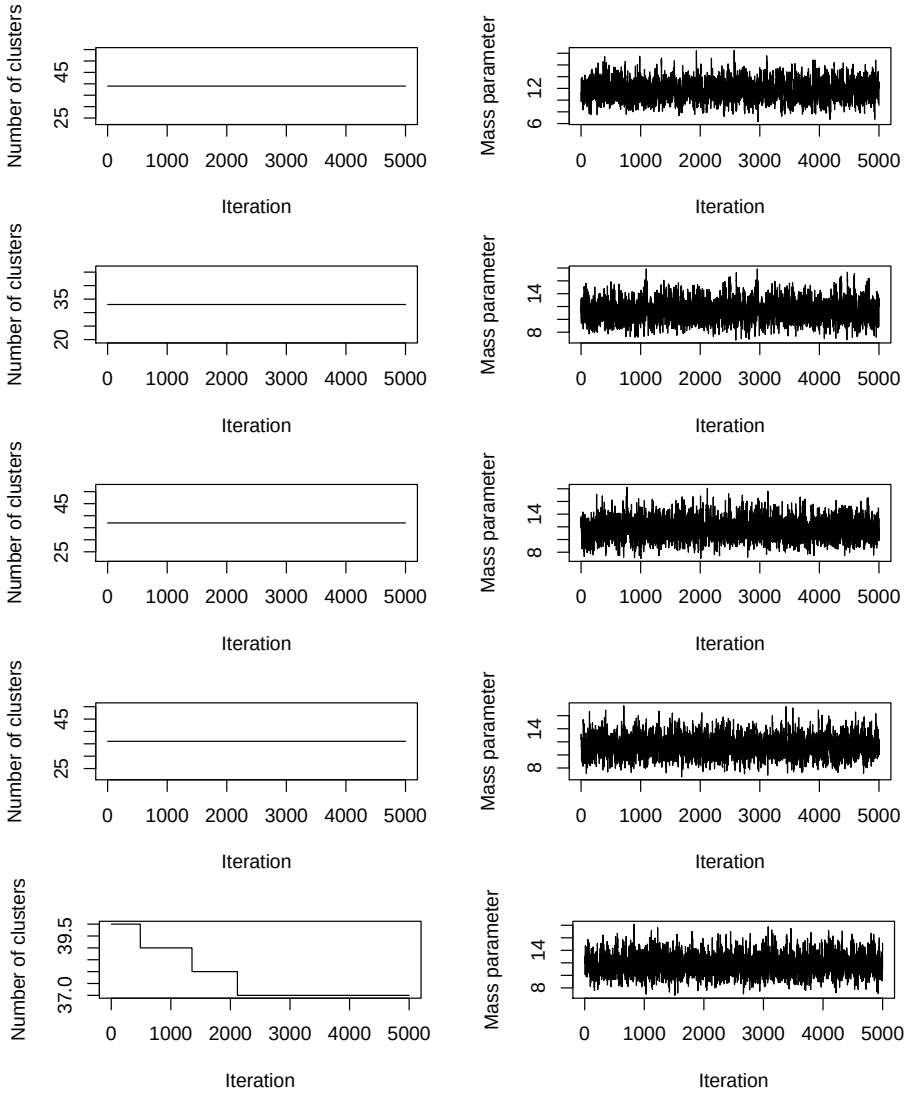


FIGURE D.18: MCMC convergence assessment, ChIP-seq data (monocytes).

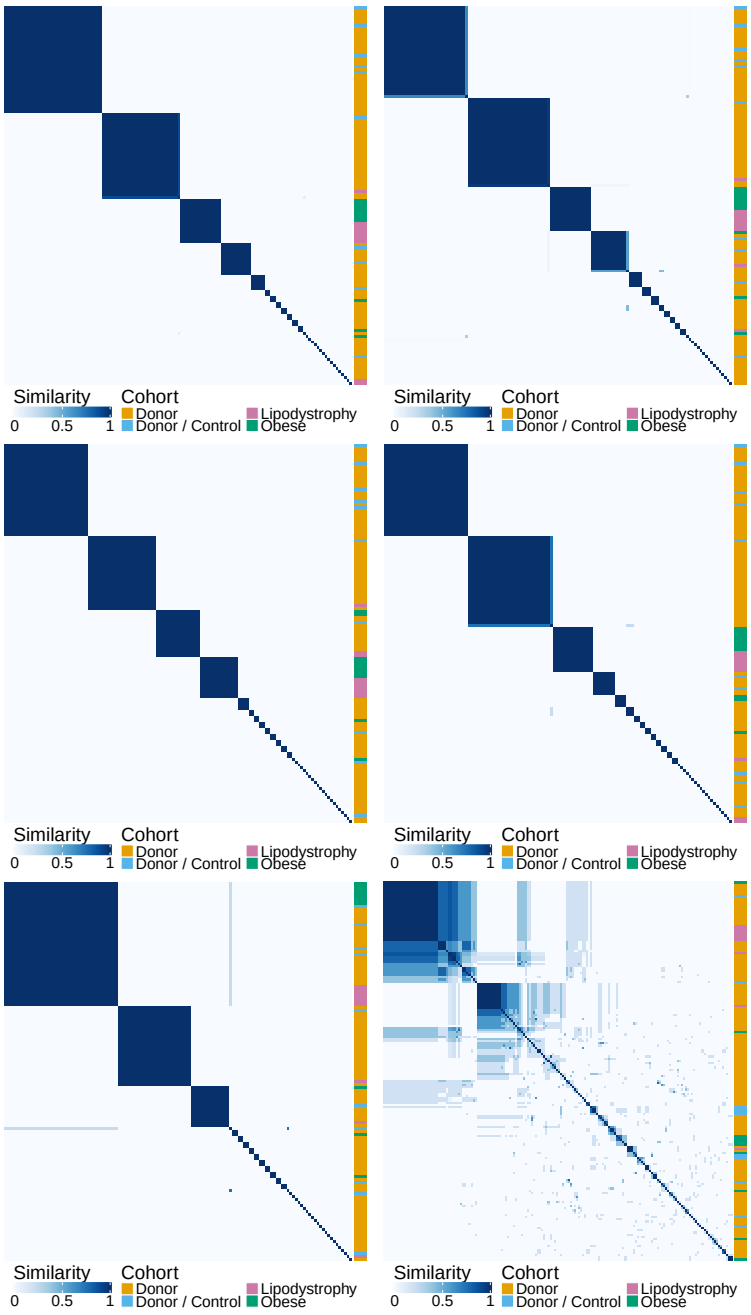


FIGURE D.19: Five PSMs of the ChIP-seq data (neutrophils) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.43	0.65	0.43	0.20
Chain 2	1	0.32	0.51	0.36
Chain 3		1	0.37	0.20
Chain 4			1	0.27

TABLE D.10: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

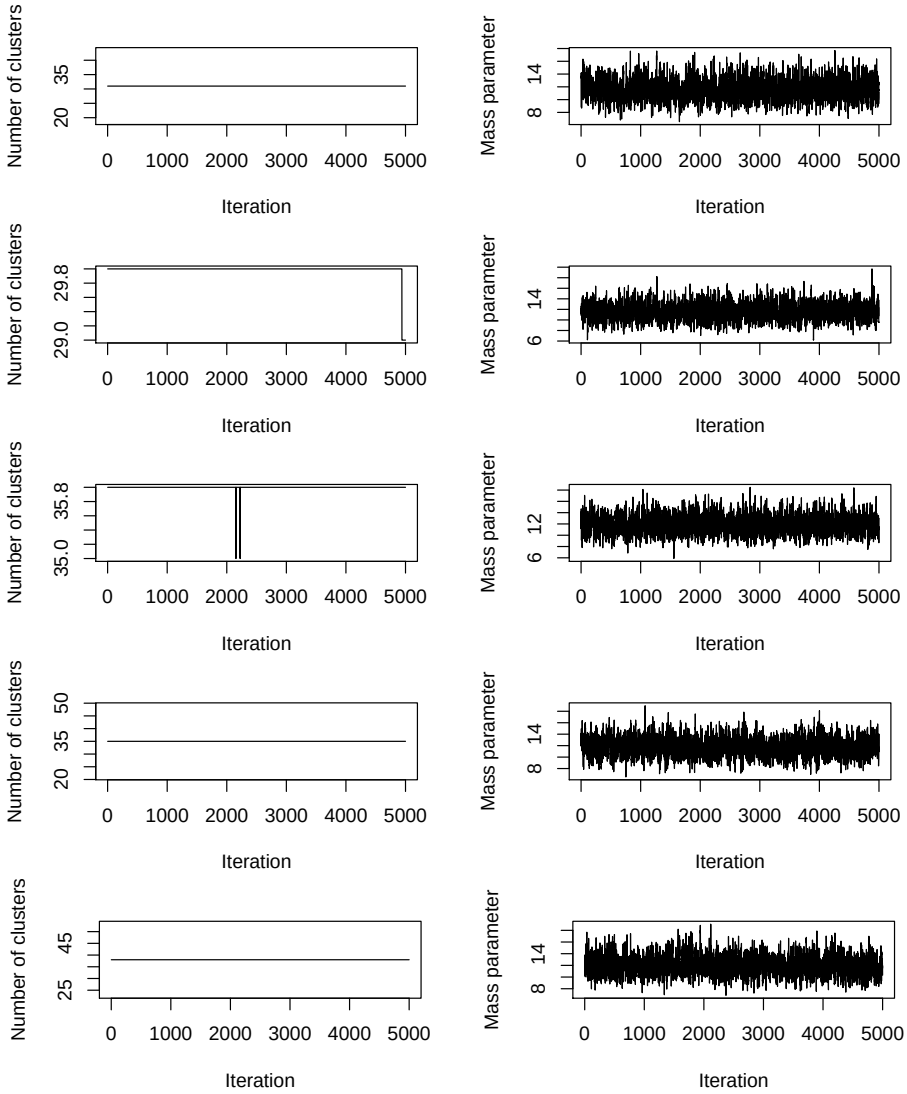


FIGURE D.20: MCMC convergence assessment, ChIP-seq data (neutrophils).

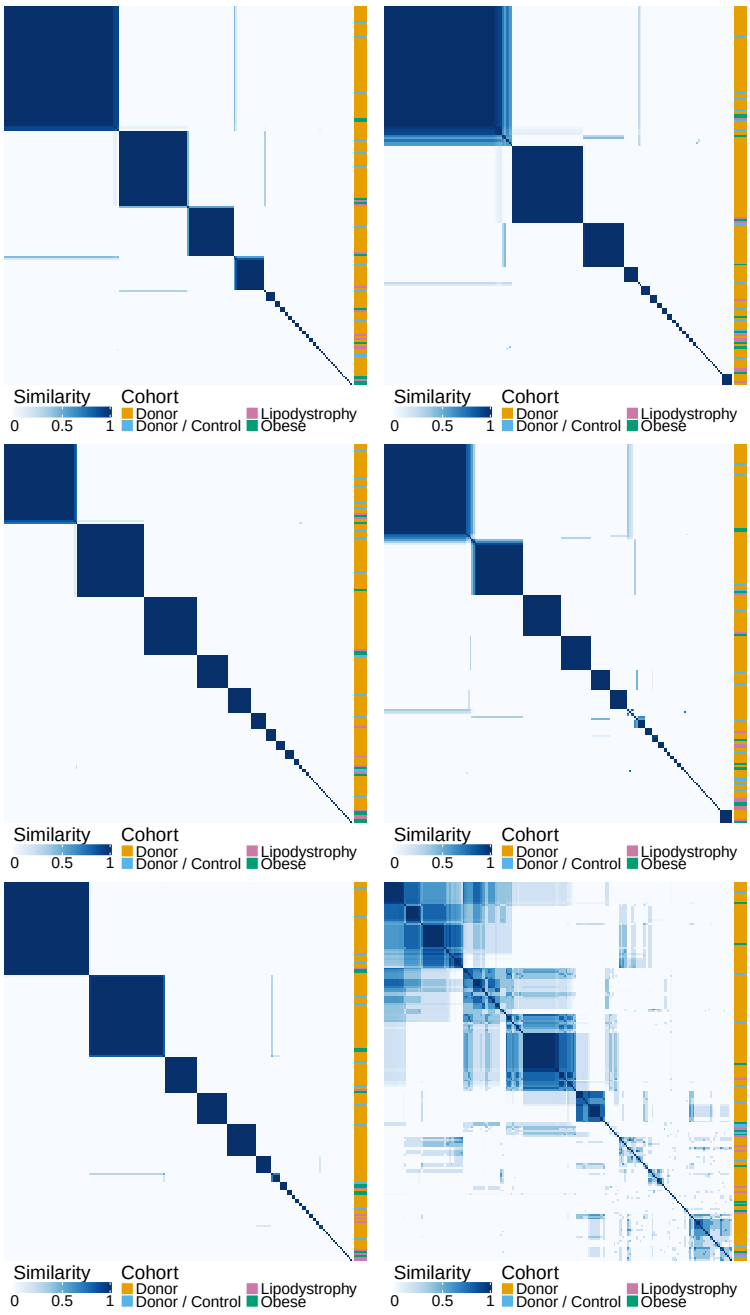


FIGURE D.21: Five PSMs of the RNA-seq (monocytes) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.21	0.47	0.38	0.34
Chain 2	1	0.33	0.25	0.42
Chain 3		1	0.35	0.45
Chain 4			1	0.49

TABLE D.11: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

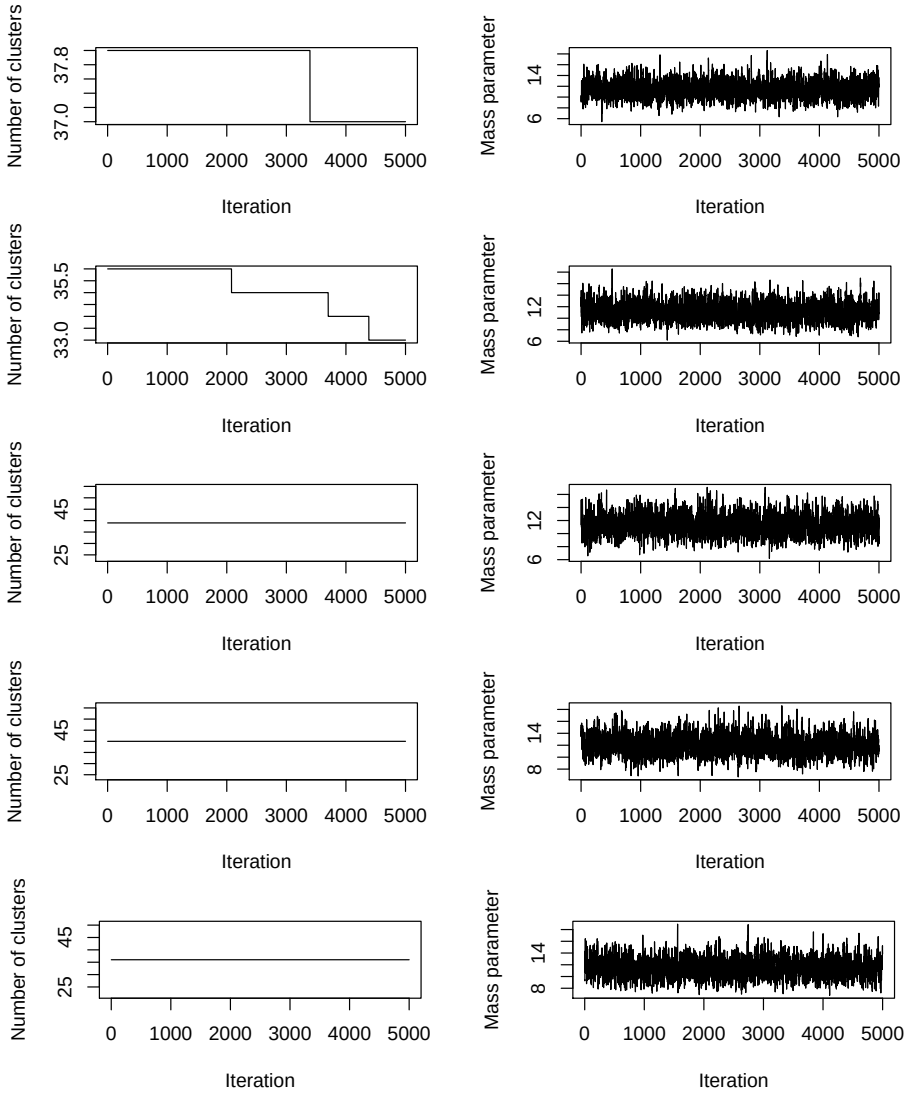


FIGURE D.22: MCMC convergence assessment, RNA-seq data (monocytes).

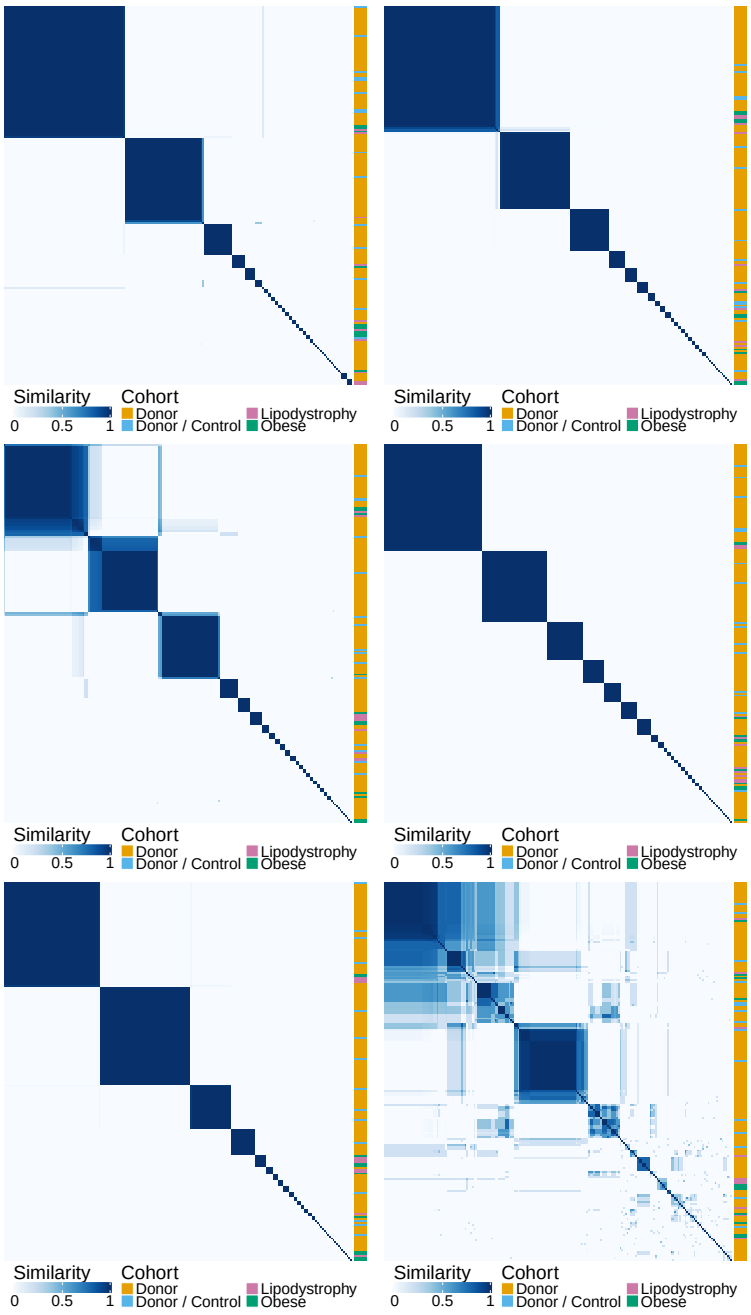


FIGURE D.23: Five PSMs of the RNA-seq (neutrophils) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.71	0.39	0.37	0.45
Chain 2	1	0.36	0.36	0.48
Chain 3		1	0.42	0.23
Chain 4			1	0.31

TABLE D.12: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

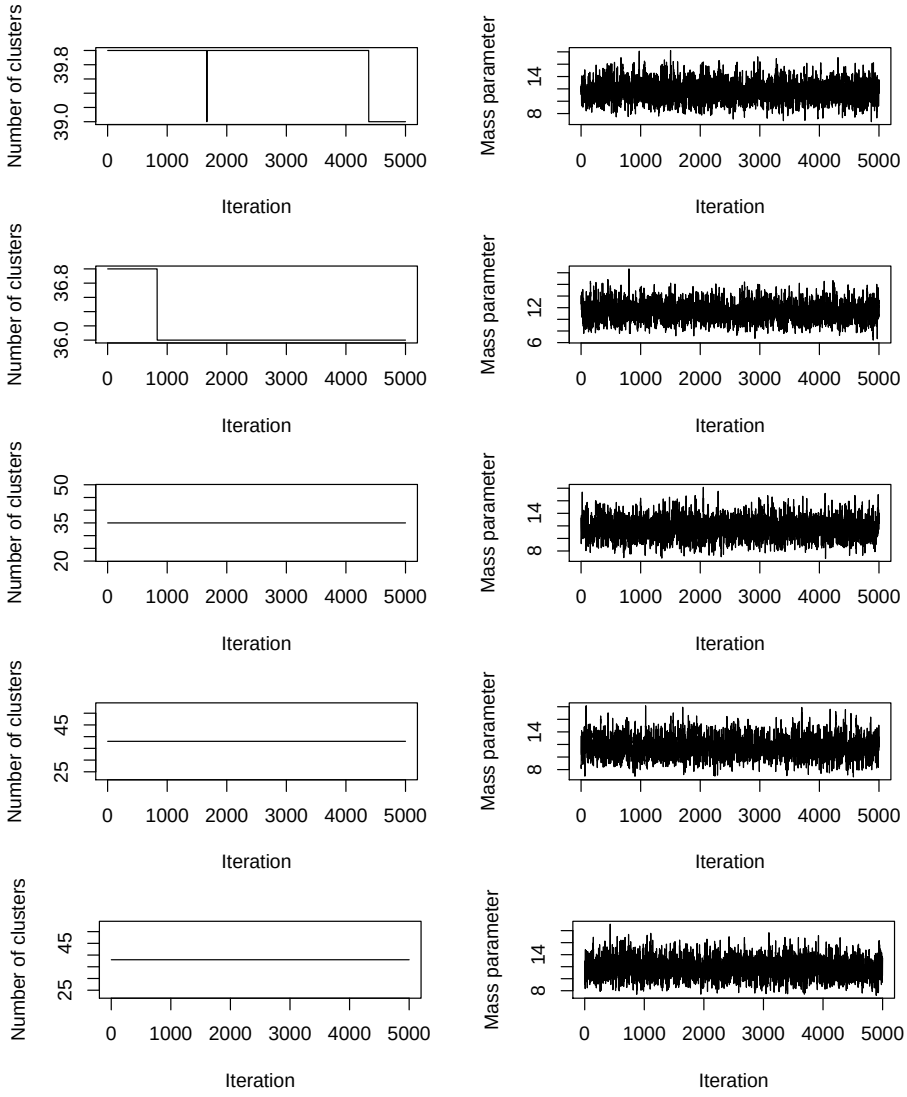


FIGURE D.24: MCMC convergence assessment, RNA-seq data (neutrophils).

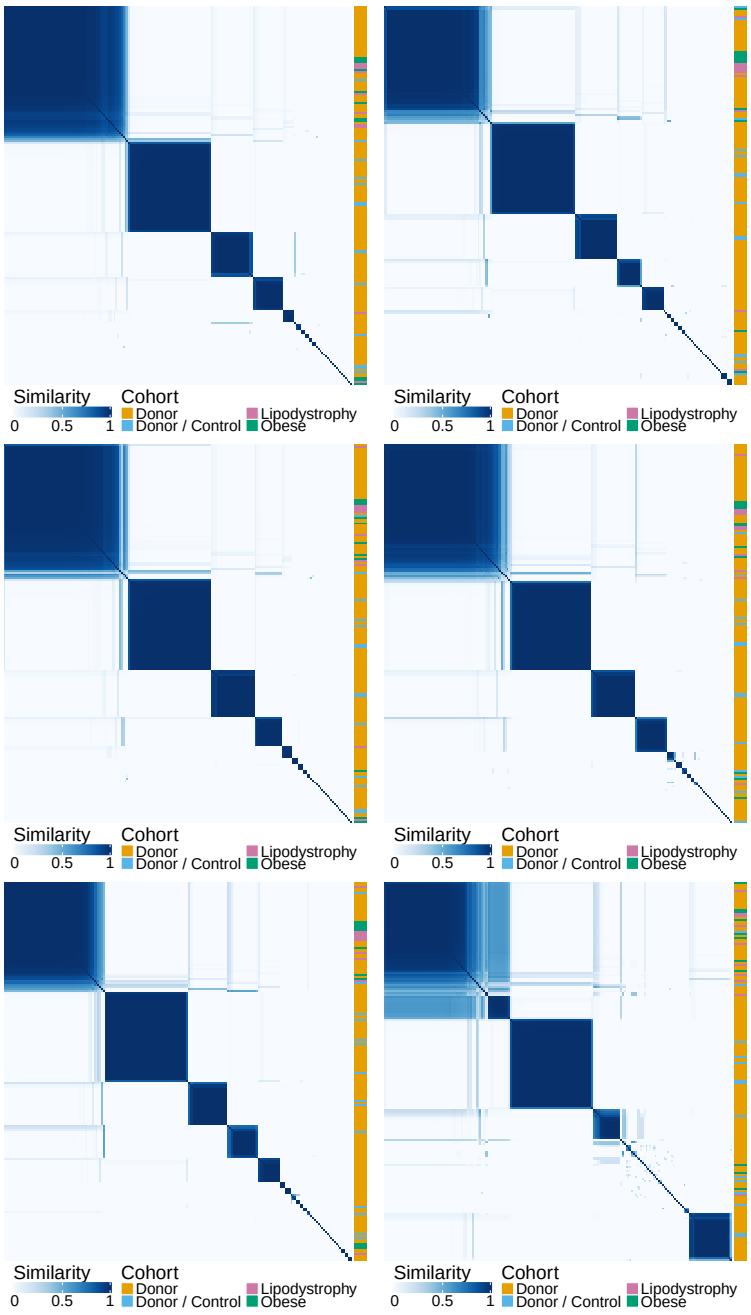


FIGURE D.25: Five PSMs of the methylation (monocytes) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.75	0.88	0.88	0.79
Chain 2	1	0.71	0.72	0.90
Chain 3		1	0.89	0.76
Chain 4			1	0.77

TABLE D.13: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

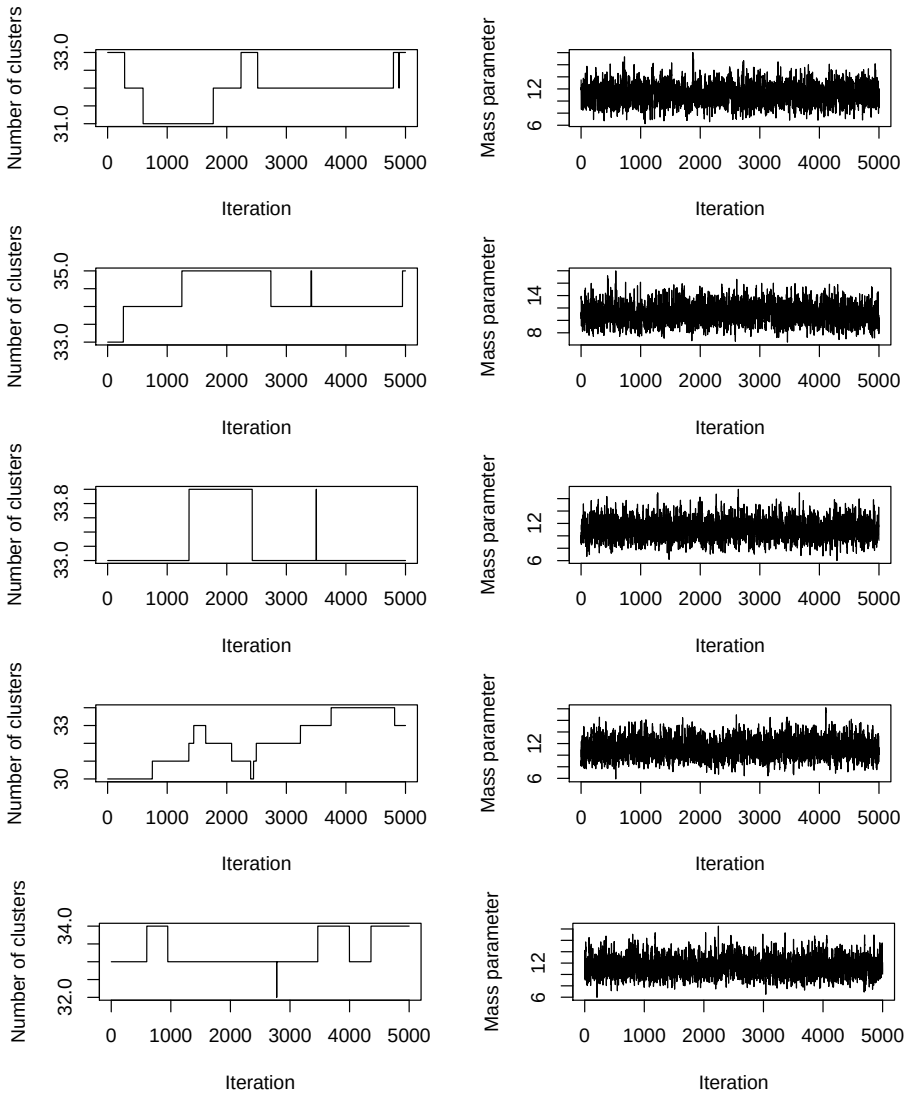


FIGURE D.26: MCMC convergence assessment, methylation data (monocytes).

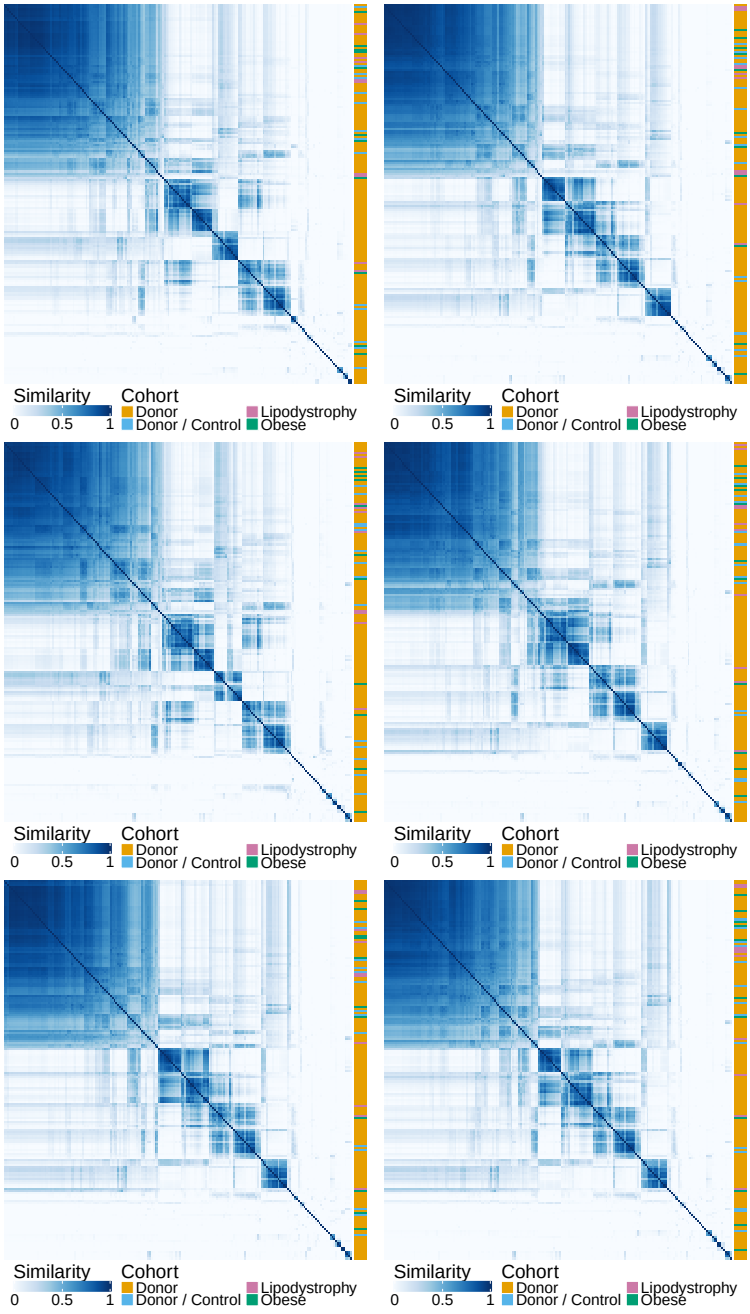


FIGURE D.27: Five PSMs of the methylation (neutrophils) and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.84	0.80	0.90	0.80
Chain 2	1	0.85	0.84	0.83
Chain 3		1	0.80	0.75
Chain 4			1	0.79

TABLE D.14: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

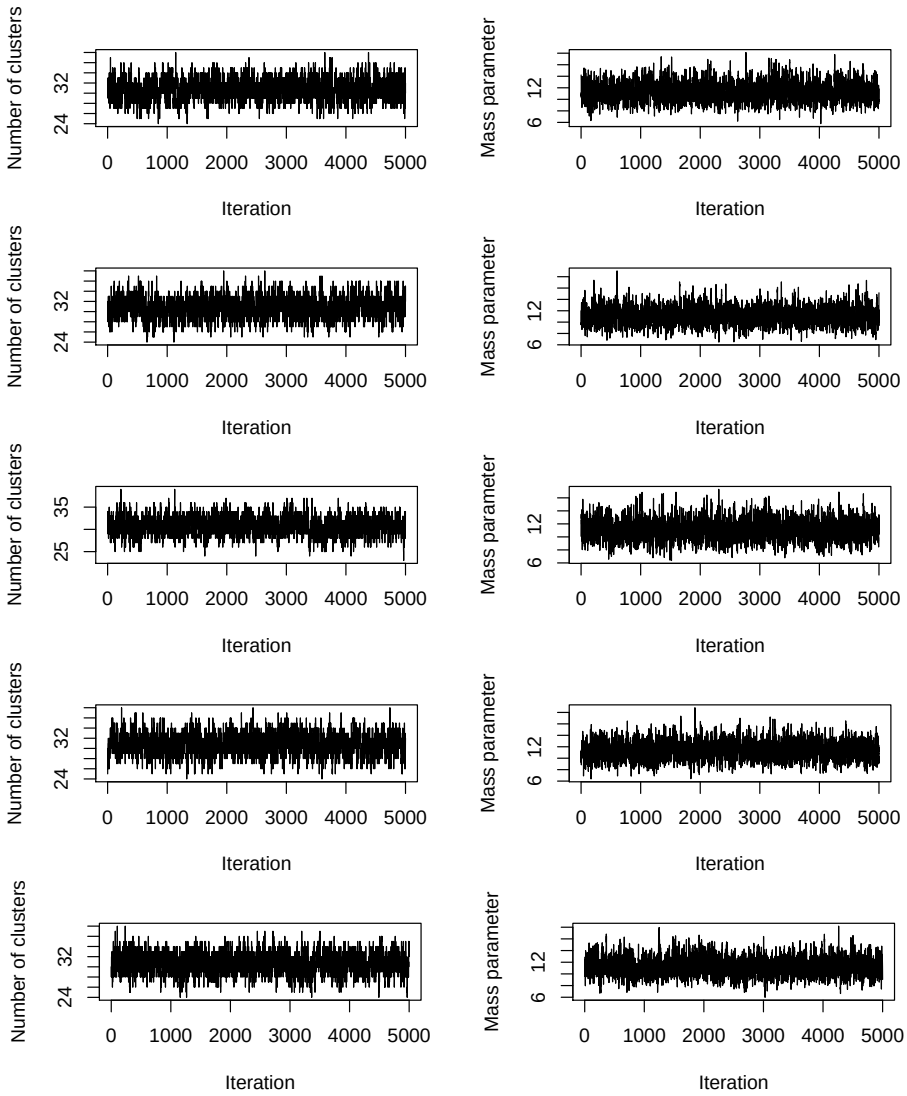


FIGURE D.28: MCMC convergence assessment, methylation data (neutrophils).

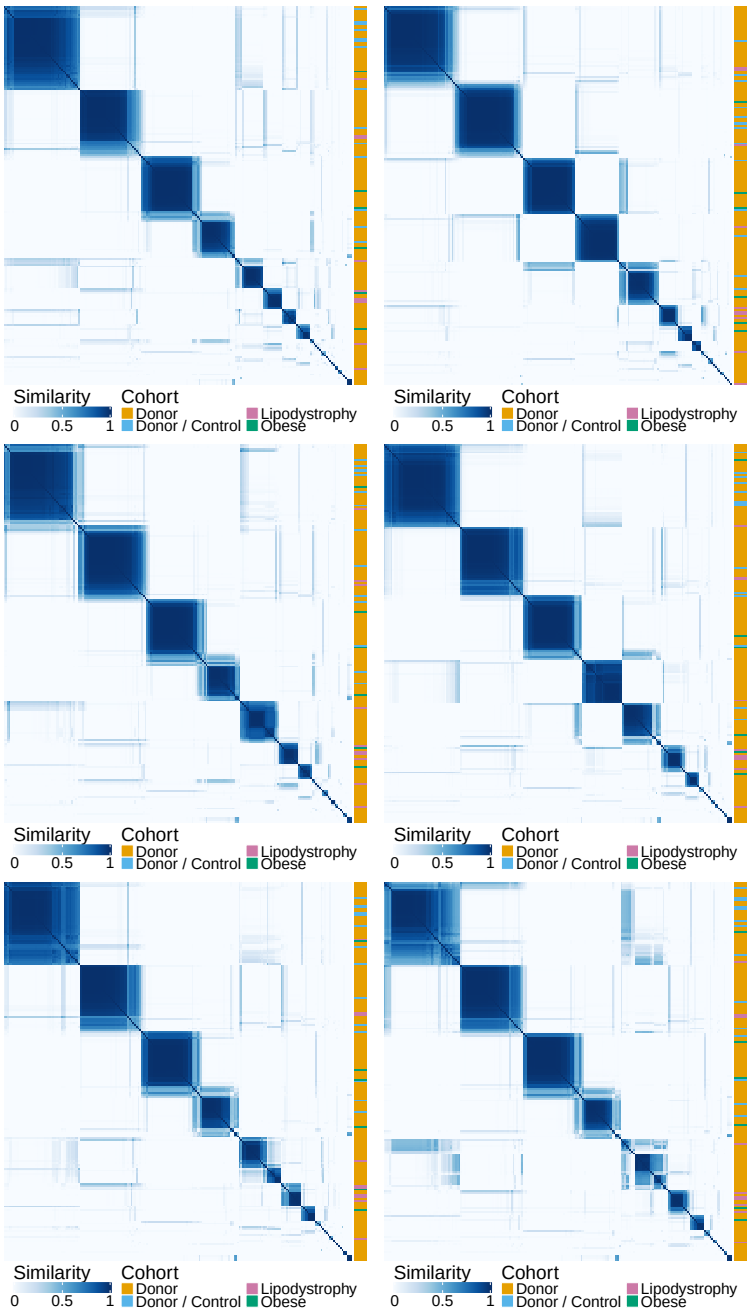


FIGURE D.29: Five PSMs of the metabolites and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.71	0.80	0.80	0.79
Chain 2	1	0.73	0.70	0.65
Chain 3		1	0.75	0.76
Chain 4			1	0.80

TABLE D.15: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

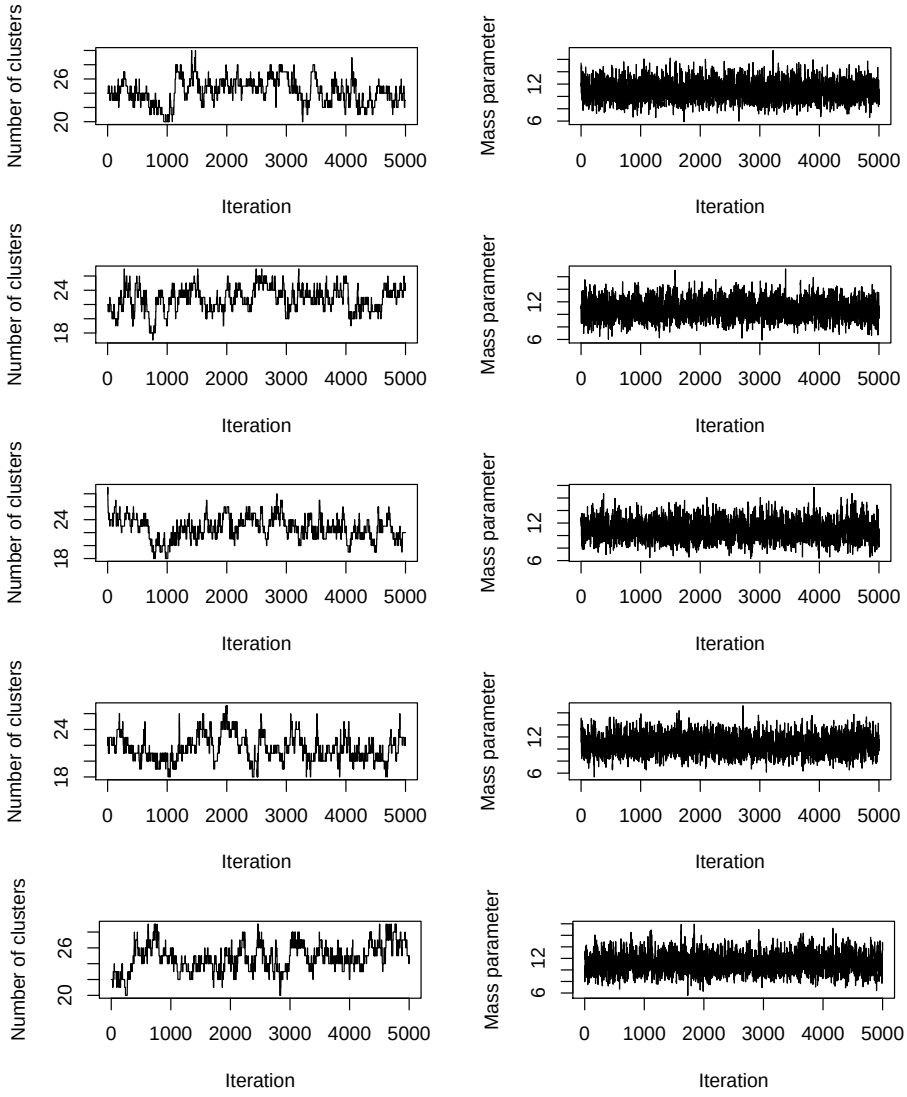


FIGURE D.30: MCMC convergence assessment, metabolite data.

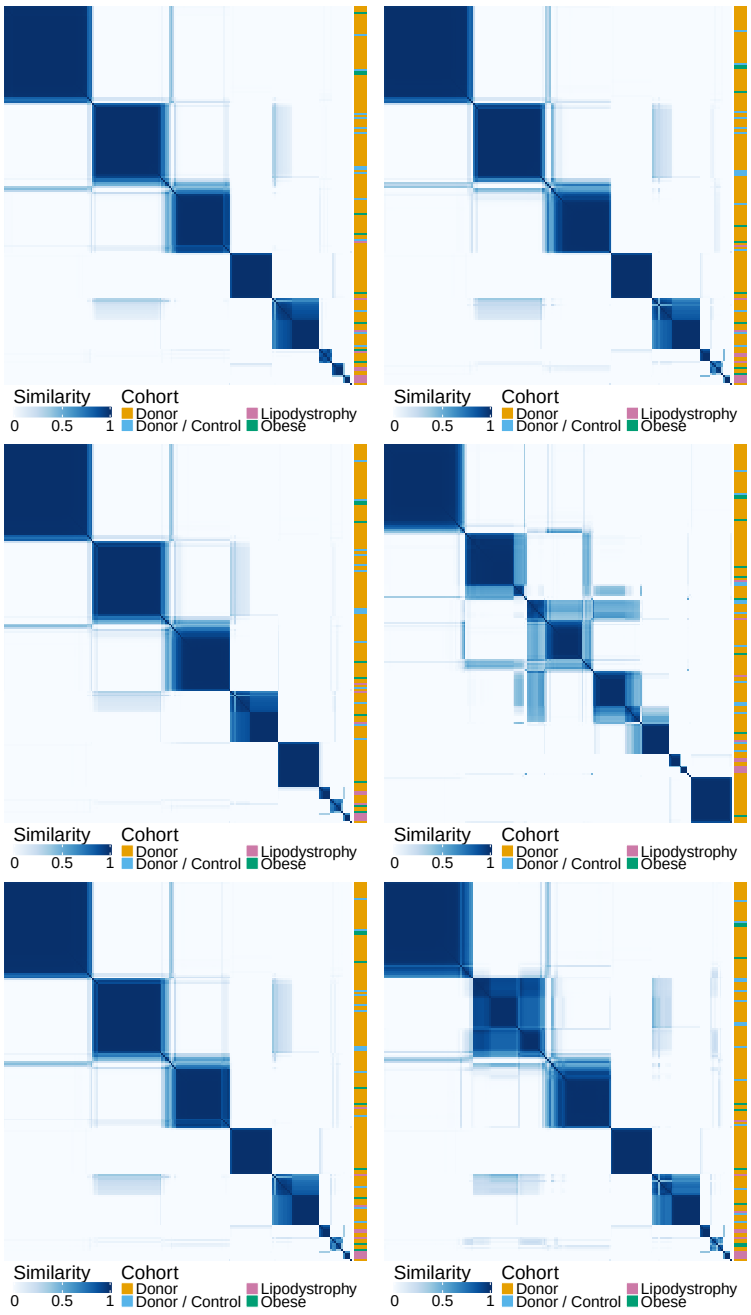


FIGURE D.31: Five PSMs of the lipids and their average (bottom right). On the right of each PSM is indicated the cohort to which each individual belongs.

	Chain 2	Chain 3	Chain 4	Chain 5
Chain 1	0.97	1.00	0.73	0.97
Chain 2	1	0.97	0.72	0.99
Chain 3		1	0.73	0.97
Chain 4			1	0.71

TABLE D.16: ARI between the clusterings found on the PSMs of different chains with the number of clusters that maximises the silhouette.

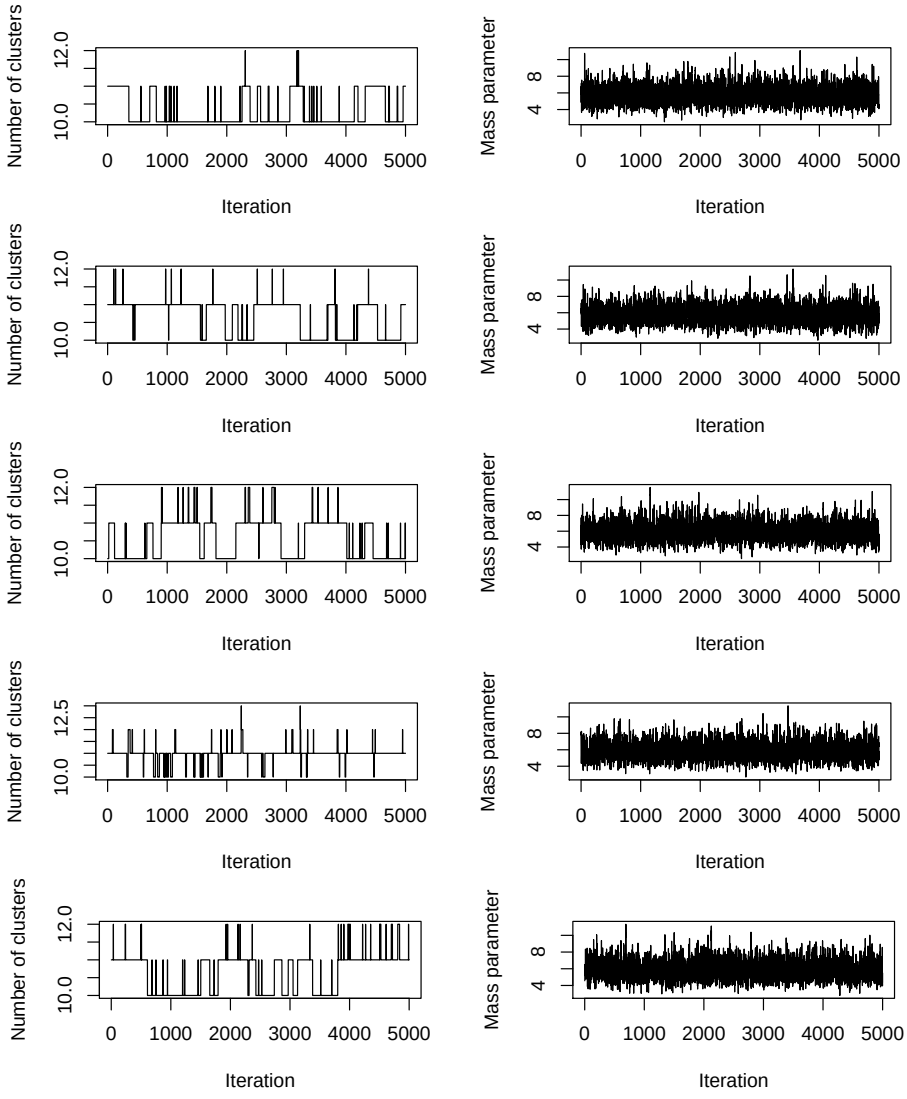


FIGURE D.32: MCMC convergence assessment, lipid data.

D.1.3 *Unsupervised integration: additional figures*

Clustering structure in the data

Figures D.33 to D.40 show how the eight data layers where the rows have been sorted by final cluster.

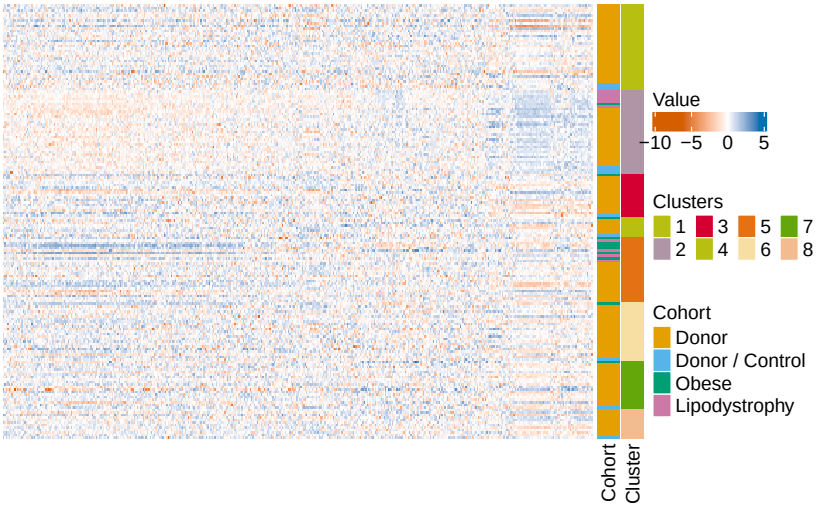


FIGURE D.33: Left: ChIP-seq data, monocytes (each row is an individual). Right: cohorts and final clusters.

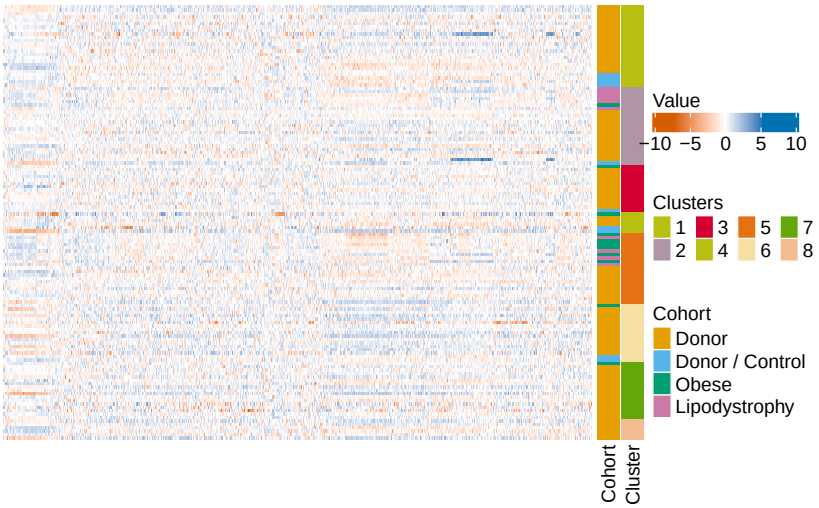


FIGURE D.34: Left: ChIP-seq data, monocytes (each row is an individual). Right: cohorts and final clusters.

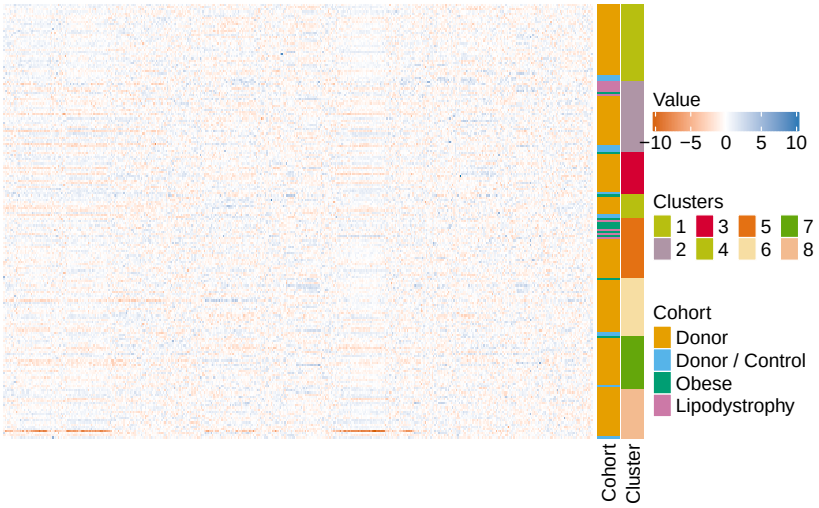


FIGURE D.35: Left: RNA-seq data, monocytes (each row is an individual). Right: cohorts and final clusters.

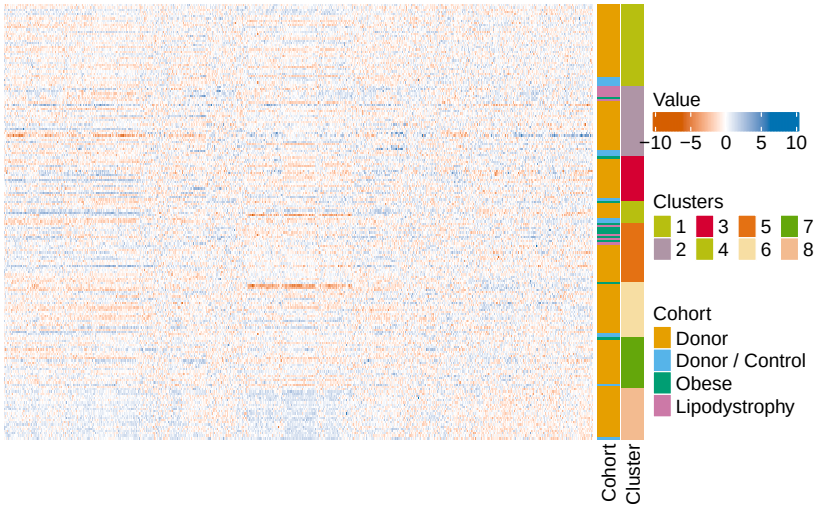


FIGURE D.36: Left: RNA-seq data, neutrophils (each row is an individual). Right: cohorts and final clusters.

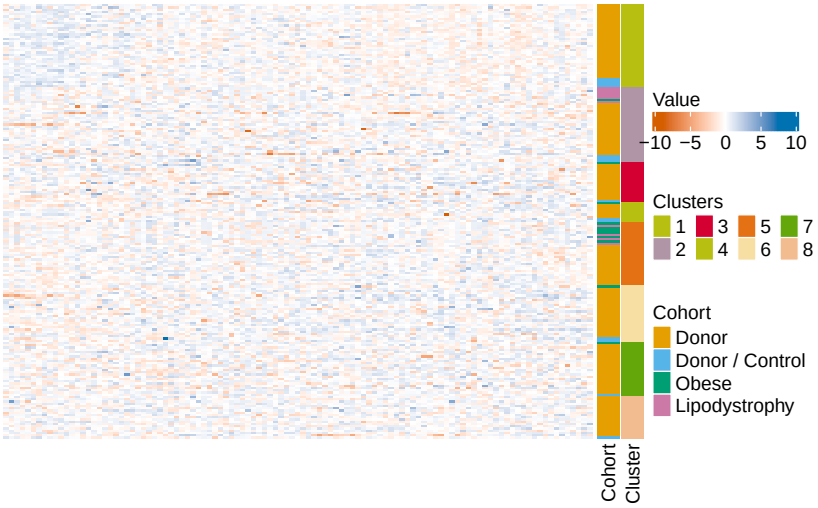


FIGURE D.37: Left: methylation data, monocytes (each row is an individual). Right: cohorts and final clusters.

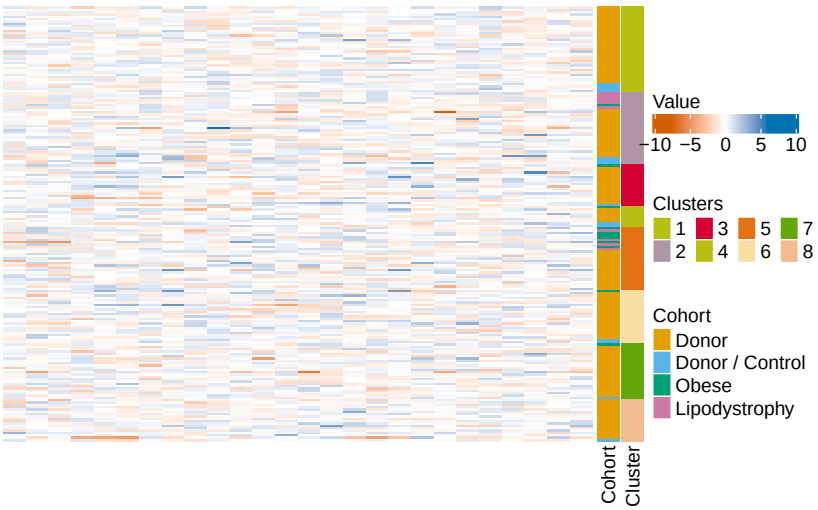


FIGURE D.38: Left: methylation data, neutrophils (each row is an individual). Right: cohorts and final clusters.

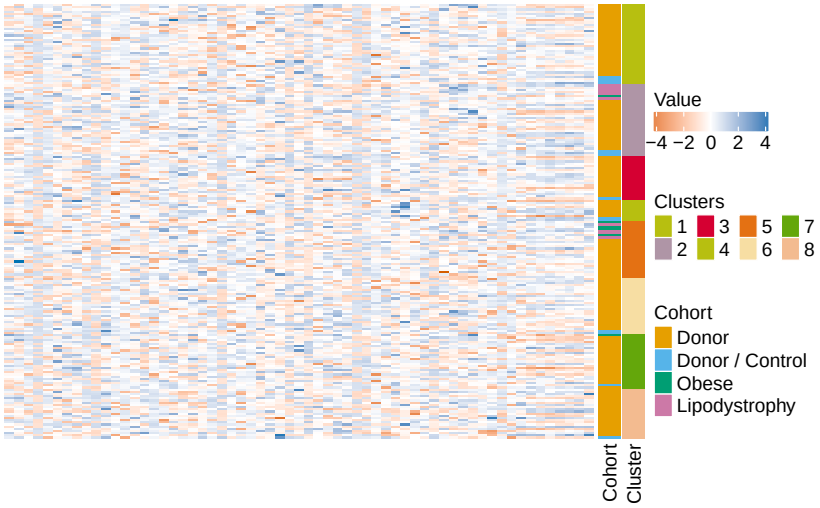


FIGURE D.39: Left: metabolite data (each row is an individual). Right: cohorts and final clusters.

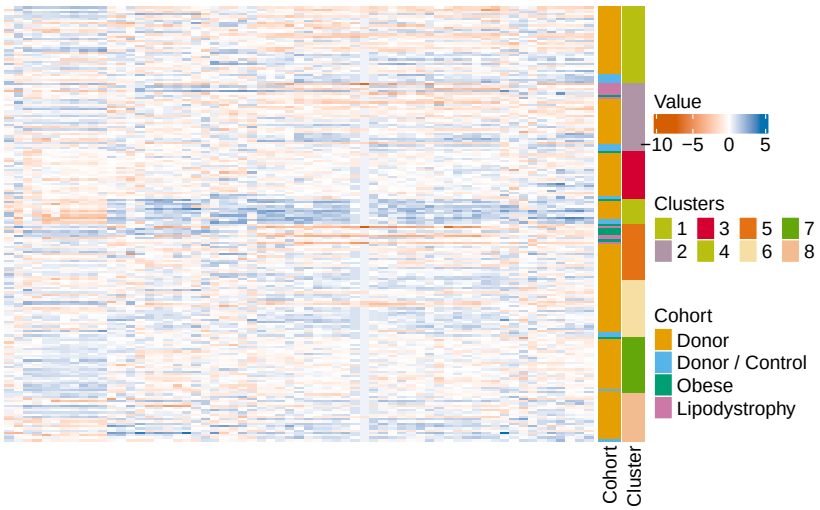


FIGURE D.40: Left: lipid data (each row is an individual). Right: cohorts and final clusters.

D.1.4 *Outcome-guided integration: additional figures*

Figures D.41 to D.48 show how the eight data layers where the rows have been sorted by final cluster.

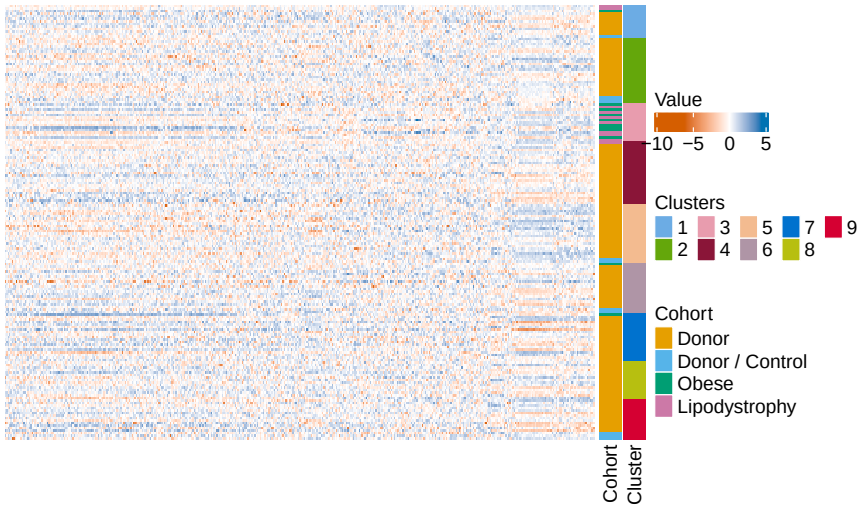


FIGURE D.41: Left: ChIP-seq data, monocytes (each row is an individual). Right: cohorts and final clusters.

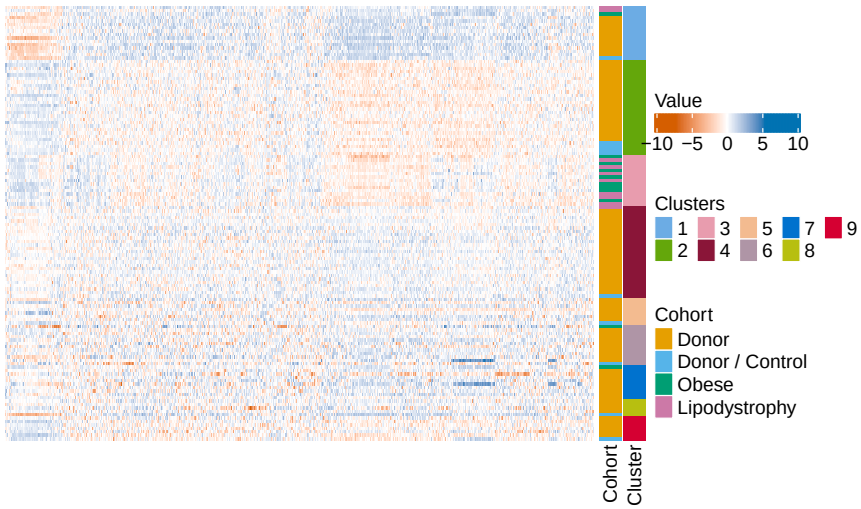


FIGURE D.42: Left: ChIP-seq data, monocytes (each row is an individual). Right: cohorts and final clusters.

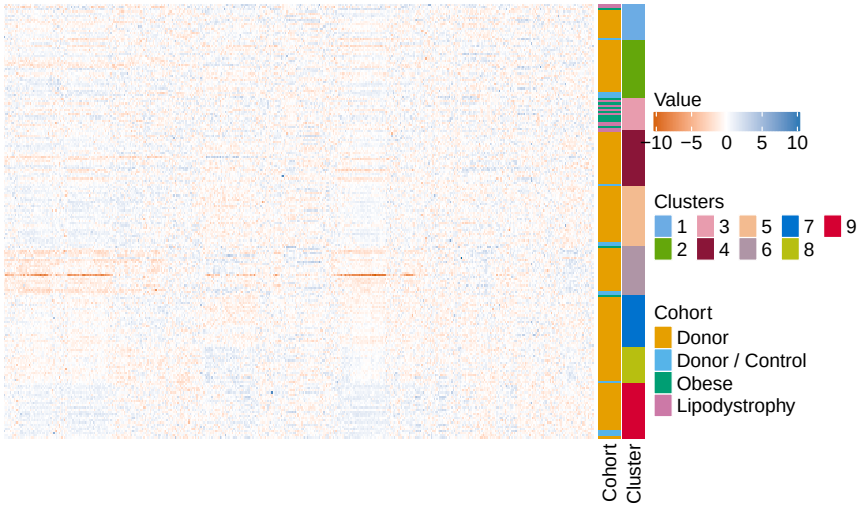


FIGURE D.43: Left: RNA-seq data, monocytes (each row is an individual). Right: cohorts and final clusters.

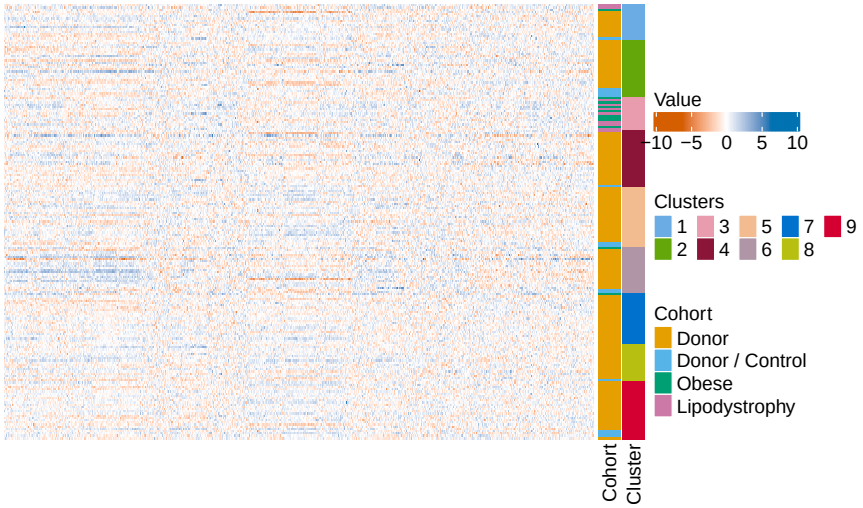


FIGURE D.44: Left: RNA-seq data, neutrophils (each row is an individual). Right: cohorts and final clusters.

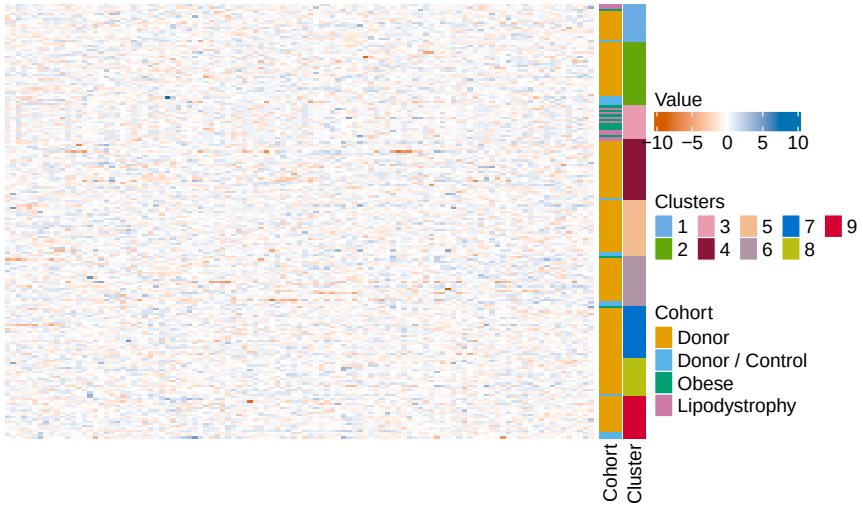


FIGURE D.45: Left: methylation data, monocytes (each row is an individual). Right: cohorts and final clusters.

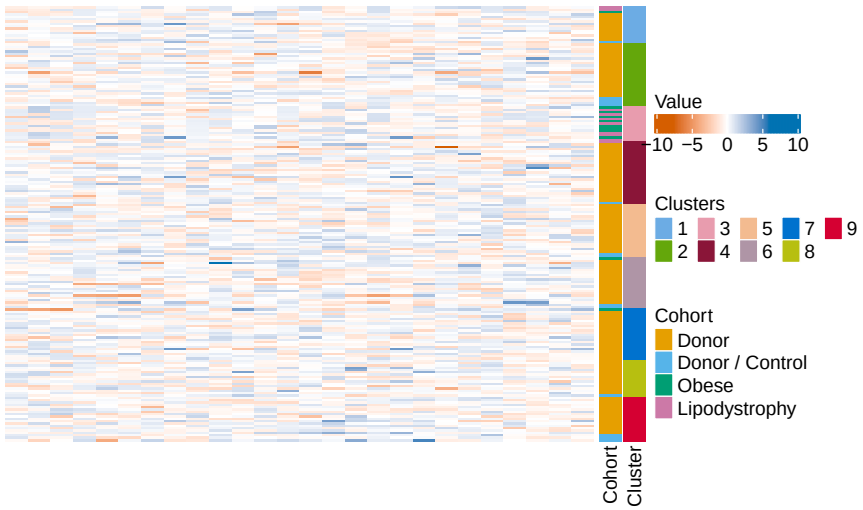


FIGURE D.46: Left: methylation data, neutrophils (each row is an individual). Right: cohorts and final clusters.

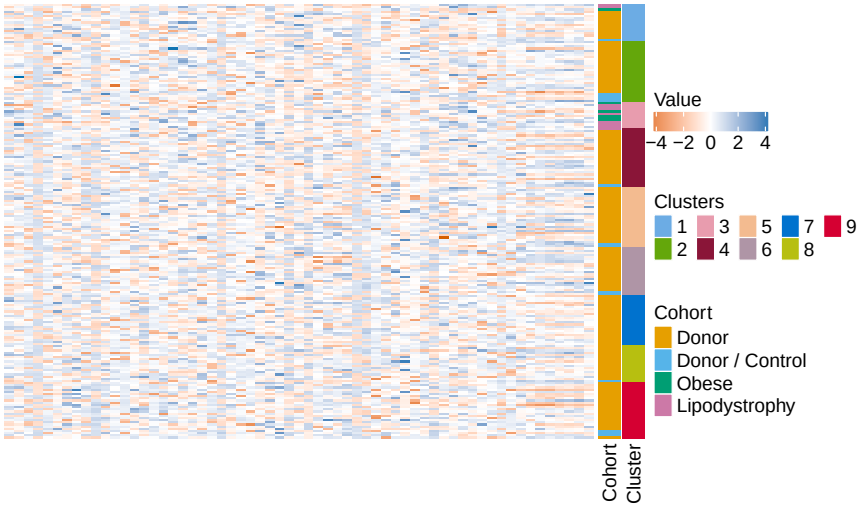


FIGURE D.47: Left: metabolite data (each row is an individual). Right: cohorts and final clusters.

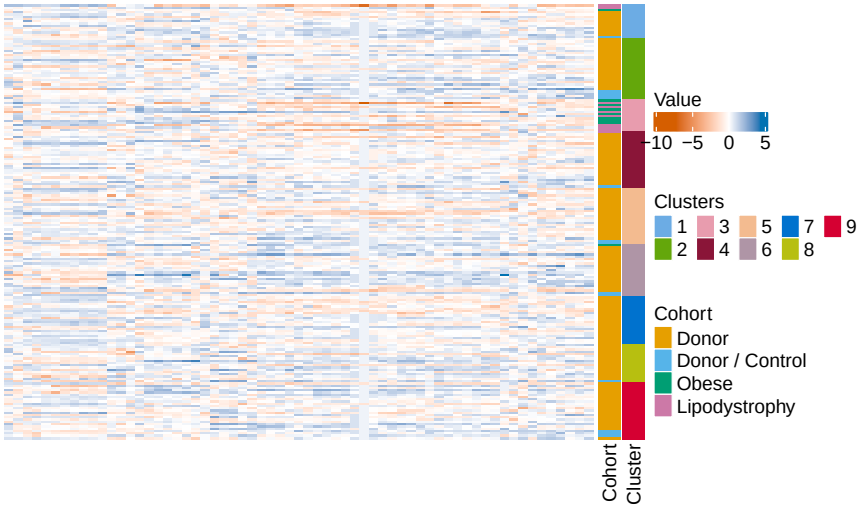


FIGURE D.48: Left: lipid data (each row is an individual). Right: cohorts and final clusters.

BIBLIOGRAPHY

Aerts, Stein et al. (2006). "Gene prioritization through genomic data fusion". In: *Nature Biotechnology* 24.5, pp. 537–544.
DOI: 10.1038/nbt1203.

Afrin, Kahkashan et al. (2020). *Directionally Dependent Multi-View Clustering Using Copula Model*.
arXiv: 2003.07494.

Ahmad, Ashar and Holger Fröhlich (2017). "Towards clinically more relevant dissection of patient heterogeneity via survival-based Bayesian clustering". In: *Bioinformatics* 33.22, pp. 3558–3566.
DOI: 10.1093/bioinformatics/btx464.

Aldous, David J (1985). "Exchangeability and related topics". In: *École d'Été de Probabilités de Saint-Flour XIII—1983*. Springer, pp. 1–198.
DOI: 10.1007/BFb0099421.

Allison, David B et al. (2006). "Microarray data analysis: from disarray to consolidation and consensus". In: *Nature Reviews Genetics* 7.1, pp. 55–65.
DOI: 10.1038/nrg1749.

Argelaguet, Ricard et al. (2018). "Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets". In: *Molecular systems biology* 14.6, e8124.
DOI: 10.15252/msb.20178124.

Arora, Arshi et al. (2020). "Pan-cancer identification of clinically relevant genomic subtypes using outcome-weighted integrative clustering". In: *bioRxiv*.
DOI: 10.1101/2020.05.11.084798.

Aure, Miriam Ragle et al. (2017). "Integrative clustering reveals a novel split in the luminal A subtype of breast cancer with impact on outcome". In: *Breast Cancer Research* 19.1, p. 44.
DOI: 10.1186/s13058-017-0812-y.

Bach, Francis R, Gert RG Lanckriet, and Michael I Jordan (2004). "Multiple kernel learning, conic duality, and the SMO algorithm". In: *Proceedings of the 21st International Conference on Machine Learning*, p. 6.
DOI: 10.1145/1015330.1015424.

Bair, Eric and Robert Tibshirani (2004). "Semi-supervised methods to predict patient survival from gene expression data". In: *PLoS Biology* 2.4, e108.
DOI: 10.1371/journal.pbio.0020108.

Bartel, David P (2004). "MicroRNAs: genomics, biogenesis, mechanism, and function". In: *Cell* 116.2, pp. 281–297.
DOI: 10.1016/S0092-8674(04)00045-5.

Baudat, Gaston and Fatiha Anouar (2000). "Generalized discriminant analysis using a kernel approach". In: *Neural Computation* 12.10, pp. 2385–2404.
DOI: 10.1162/089976600300014980.

Bellet, Aurélien, Amaury Habrard, and Marc Sebban (2013). *A survey on metric learning for feature vectors and structured data*.
arXiv: 1306.6709.

Bennett, Kristin and Olvi L Mangasarian (1992). "Robust linear programming discrimination of two linearly inseparable sets". In: *Optimization Methods and Software* 1.1, pp. 23–34.
DOI: 10.1080/10556789208805504.

Bersanelli, Matteo et al. (2016). "Methods for the integration of multi-omics data: mathematical aspects". In: *BMC Bioinformatics* 17.S2, S15.
DOI: 10.1186/s12859-015-0857-9.

Binder, David A (1978). "Bayesian cluster analysis". In: *Biometrika* 1, pp. 31–38.
DOI: 10.1093/biomet/65.1.31.

Bishop, Christopher M (2006). *Pattern Recognition and Machine Learning*. 1st edition. Springer.
URL: https://www.springer.com/gb/book/9780387310732.

Blangiardo, Marta and Sylvia Richardson (2007). "Statistical tools for synthesizing lists of differentially expressed features in related experiments". In: *Genome Biology* 8.4, R54.
DOI: 10.1186/gb-2007-8-4-r54.

Blausen.com staff (2014). "Medical gallery of Blausen Medical 2014". In: *Wiki Journal of Medicine* 1.2.
DOI: 10.15347/wjm/2014.010.

Blei, David M, Michael I Jordan, et al. (2006). "Variational inference for Dirichlet process mixtures". In: *Bayesian Analysis* 1.1, pp. 121–143.
DOI: 10.1214/06-BA104.

Bonferroni, Carlo (1936). "Teoria statistica delle classi e calcolo delle probabilità". In: *Pubblicazioni dell'Istituto Superiore di Scienze Economiche e Commericiali di Firenze* 8, pp. 3–62.

Boser, Bernhard E, Isabelle M Guyon, and Vladimir N Vapnik (1992). "A training algorithm for optimal margin classifiers". In: *Proceedings of the 5th annual workshop on Computational Learning Theory*.
DOI: 10.1145/130385.130401.

Boulesteix, Anne-Laure et al. (2017). "IPF-LASSO: Integrative-penalized regression with penalty factors for prediction based on multi-omics data". In: *Computational and Mathematical Methods in Medicine* 2017.
DOI: 10.1155/2017/7691937.

Brunet, Jean-Philippe et al. (2004). "Metagenes and molecular pattern discovery using matrix factorization". In: *Proceedings of the National Academy of Sciences* 101.12, pp. 4164–4169.
DOI: 10.1073/pnas.0308531101.

Cabassi, Alessandra, Mehmet Gönen, and Paul DW Kirk (2020). *klic: Kernel Learning Integrative Clustering*. R package version 1.0.2.
DOI: 10.5281/zenodo.3739391.

Cabassi, Alessandra and Paul DW Kirk (2020a). *coca: Cluster-Of-Clusters Analysis*. R package version 1.0.4.
DOI: 10.5281/zenodo.3727811.

– (2020b). "Multiple kernel learning for integrative consensus clustering of 'omic datasets". In: *Bioinformatics*. btaa593.
DOI: 10.1093/bioinformatics/btaa593.

Cabassi, Alessandra, Sylvia Richardson, and Paul DW Kirk (2020). *Kernel learning approaches for summarising and combining posterior similarity matrices*.
arXiv: 2009.12852.

Cabassi, Alessandra et al. (2020). *Two-step penalised logistic regression for multi-omic data with an application to cardiometabolic syndrome*.
arXiv: 2008.00235.

Caielli, Simone, Jacques Banchereau, and Virginia Pascual (2012). "Neutrophils come of age in chronic inflammation". In: *Current Opinion in Immunology* 24.6, pp. 671–677.
DOI: 10.1016/j.coi.2012.09.008.

Calvo, Sarah et al. (2006). "Systematic identification of human mitochondrial disease genes through integrative genomics". In: *Nature genetics* 38.5, pp. 576–582.
DOI: 10.1038/ng1776.

Campos, Eric I and Danny Reinberg (2009). "Histones: annotating chromatin". In: *Annual review of genetics* 43, pp. 559–599.
DOI: 10.1146/annurev.genet.032608.103928.

Chang, Jason and John W Fisher III (2014). "Parallel sampling of HDPs using sub-cluster splits". In: *Advances in Neural Information Processing Systems*, pp. 235–243.
URL: http://papers.nips.cc/paper/5235-parallel-sampling-of-hdps-using-sub-cluster-splits.

Chaudhary, Kumardeep et al. (2018). "Deep learning–based multi-omics integration robustly predicts survival in liver cancer". In: *Clinical Cancer Research* 24.6, pp. 1248–1259.
DOI: 10.1158/1078-0432.CCR-17-0853.

Chen, Yihua et al. (2009). "Similarity-based classification: Concepts and algorithms." In: *Journal of Machine Learning Research* 10.3.
URL: https://jmlr.csail.mit.edu/papers/v10/chen09a.html.

Chung, Yeonseung and David B Dunson (2009). "Nonparametric Bayes conditional distribution modeling with variable selection". In: *Journal of the American Statistical Association* 104.488, pp. 1646–1660.
DOI: 10.1198/jasa.2009.tm08302.

Cirulli, Elizabeth T et al. (2019). "Profound perturbation of the metabolome in obesity is associated with health risk". In: *Cell Metabolism* 29.2, pp. 488–500.
DOI: 10.1016/j.cmet.2018.09.022.

Cooke, Emma J et al. (2011). "Bayesian hierarchical clustering for microarray time series data with replicates and outlier measurements". In: *BMC Bioinformatics* 12.1, p. 399.
DOI: 10.1186/1471-2105-12-399.

Cover, Thomas M. and Joy A. Thomas (2006). *Elements of Information Theory*. Wiley Interscience.

Cox, David R (1972). "Regression models and life-tables". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2, pp. 187–202.
DOI: 10.1111/j.2517-6161.1972.tb00899.x.

Cox, Michael AA and Trevor F Cox (2008). "Multidimensional scaling". In: *Handbook of data visualization*. Springer, pp. 315–347.
DOI: 10.1007/978-3-540-33037-0_14.

Cramer, Jan Salomon (2002). *The origins of logistic regression*. Tech. rep. 119/4. Tinbergen Institute.

Crick, Francis (1970). "Central dogma of molecular biology". In: *Nature* 227.5258, pp. 561–563.
DOI: 10.1038/227561a0.

Crick, Francis HC (1958). "On protein synthesis". In: *Symposia of the Society of Experimental Biology*. Vol. 12. 138-63, p. 8.

Crook, Oliver M, Laurent Gatto, and Paul DW Kirk (2019). "Fast approximate inference for variable selection in Dirichlet process mixtures, with an application to pan-cancer proteomics". In: *Statistical Applications in Genetics and Molecular Biology* 18.6.
DOI: 10.1515/sagmb-2018-0065.

Csató, Lehel and Manfred Opper (2002). "Sparse on-line Gaussian processes". In: *Neural Computation* 14.3, pp. 641–668.
DOI: 10.1162/089976602317250933.

Cutler, Roy G et al. (2004). "Involvement of oxidative stress-induced abnormalities in ceramide and cholesterol metabolism in brain aging and Alzheimer's disease". In: *Proceedings of the National Academy of Sciences* 101.7, pp. 2070–2075.
DOI: 10.1073/pnas.0305799101.

Dahl, David B (2006). "Model-based clustering for expression data via a Dirichlet process mixture model". In: *Bayesian inference for gene expression and proteomics*.

Ed. by Kim-Anh Do, Peter Müller and Marina Vannucci. Cambridge University Press, pp. 201–218.
DOI: 10.1093/bib/bbm022.

Dai, Jiarong et al. (2019). "Serum 3-carboxy-4-methyl-5-propyl-2-furanpropanoic acid is associated with lipid profiles and might protect against non-alcoholic fatty liver disease in Chinese individuals". In: *Journal of Diabetes Investigation* 10.3, pp. 793–800.
DOI: 10.1111/jdi.12963.

de Borda, Jean-Charles (1781). "Mémoire sur les élections au scrutin". In: *Histoire de l'Academie Royale des Sciences*.

de Gusmao Correia, Marcelo L and William G Haynes (2004). "Leptin, obesity and cardiovascular disease". In: *Current Opinion in Nephrology and Hypertension* 13.2, pp. 215–223.
DOI: 10.1097/00041552-200403000-00010.

De Hoon, Michiel JL et al. (2004). "Open source clustering software". In: *Bioinformatics* 20.9, pp. 1453–1454.
DOI: 10.1093/bioinformatics/bth078.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22.
DOI: 10.1111/j.2517-6161.1977.tb01600.x.

Dudoit, Sandrine and Jane Fridlyand (2002). "A prediction-based resampling method for estimating the number of clusters in a dataset". In: *Genome Biology* 3.7.
DOI: 10.1186/gb-2002-3-7-research0036.

Dunn, Joseph C (1974). "Well-separated clusters and optimal fuzzy partitions". In: *Journal of Cybernetics* 4.1, pp. 95–104.
DOI: 10.1080/01969727408546059.

Egert, Markus et al. (2006). "Beyond diversity: functional microbiomics of the human colon". In: *Trends in Microbiology* 14.2, pp. 86–91.
DOI: 10.1016/j.tim.2005.12.007.

Everitt, Brian S (1993). *Cluster analysis*. Hodder Education.

Fazzari, Melissa J and John M Greally (2004). "Epigenomics: beyond CpG islands". In: *Nature Reviews Genetics* 5.6, pp. 446–455.

Ferguson, Thomas S (1973). "A Bayesian Analysis of some nonparametric problems". In: *The Annals of Statistics*, pp. 209–230.

Fiorenza, Christina G, Sharon H Chou, and Christos S Mantzoros (2011). "Lipodystrophy: pathophysiology and advances in treatment". In: *Nature Reviews Endocrinology* 7.3, p. 137.
DOI: 10.1038/nrendo.2010.199.

Fop, Michael and T Brendan Murphy (2018). "Variable selection methods for model-based clustering". In: *Statistics Surveys* 12.0, pp. 18–65. ISSN: 1935-7516.
DOI: 10.1214/18-ss119.

Fraley, Chris and Adrian E Raftery (2002). "Model-based clustering, discriminant analysis, and density estimation". In: *Journal of the American statistical Association* 97.458, pp. 611–631.
DOI: 10.1198/016214502760047131.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. 10. Springer Series in Statistics.
DOI: 10.1007/b94608.

– (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent". In: *Journal of Statistical Software* 33.1, pp. 1–22.
DOI: 10.18637/jss.v033.i01.

Fritsch, Arno and Katja Ickstadt (2009). "Improved Criteria for Clustering Based on the Posterior Similarity Matrix". In: *Bayesian Analysis* 4.2, pp. 367–392.
DOI: 10.1214/09-BA414.

Fröhlich, Holger and Andreas Zell (2005). "Efficient parameter selection for support vector machines in classification and regression via model-based global optimization". In: *Proceedings of the IEEE International Joint Conference on Neural Networks, 2005.* Vol. 3. IEEE, pp. 1431–1436.
DOI: 10.1109/IJCNN.2005.1556085.

Gabašová, Evelina, John Reid, and Lorenz Wernisch (2017). "Clusternomics: Integrative context-dependent clustering for heterogeneous datasets". In: *PLoS Computational Biology* 13.10, e1005781.
DOI: 10.1371/journal.pcbi.1005781.

Ge, Hong et al. (2015). "Distributed inference for Dirichlet process mixture models". In: *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2276–2284.
URL: http://proceedings.mlr.press/v37/gea15.html.

Gelfand, Alan E and Adrian FM Smith (1990). "Sampling-based approaches to calculating marginal densities". In: *Journal of the American Statistical Association* 85.410, pp. 398–409.
DOI: 10.1080/01621459.1990.10476213.

Girolami, Mark (2002). "Mercer kernel-based clustering in feature space". In: *IEEE Transactions on Neural Networks* 13.3, pp. 780–784.
DOI: 10.1109/TNN.2002.1000150.

Gönen, Mehmet and Ethem Alpaydın (2011). "Multiple Kernel Learning Algorithms". In: *Journal of Machine Learning Research* 12.Jul, pp. 2211–2268.
URL: http://www.jmlr.org/papers/v12/gonen11a.html.

Gönen, Mehmet and Adam A Margolin (2014). "Localized data fusion for kernel k-means clustering with application to cancer biology". In: *Advances in Neural Information Processing Systems*, pp. 1305–1313.
URL: http://papers.nips.cc/paper/5236-localized-data-fusion-for-kernel-k-means-clustering-with-application-to-cancer-biology.

Goodwin, Sara, John D McPherson, and W Richard McCombie (2016). "Coming of age: ten years of next-generation sequencing technologies". In: *Nature Reviews Genetics* 17.6, p. 333.
DOI: 10.1038/nrg.2016.49.

Gordon, Edgar S (1960). "Non-esterified fatty acids in the blood of obese and lean subjects". In: *The American Journal of Clinical Nutrition* 8.5, pp. 740–747.
DOI: 10.1093/ajcn/8.5.740.

Gower, John C (1971). "A general coefficient of similarity and some of its properties". In: *Biometrics*, pp. 857–871.
DOI: 10.2307/2528823.

Granovskaia, Marina V et al. (2010). "High-resolution transcription atlas of the mitotic cell cycle in budding yeast". In: *Genome Biology* 11.3, R24.
DOI: 10.1186/gb-2010-11-3-r24.

Green, Peter J (1995). "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination". In: *Biometrika* 82.4, pp. 711–732.
DOI: 10.1093/biomet/82.4.711.

Greenberg, SA et al. (2002). "Molecular profiles of inflammatory myopathies". In: *Neurology* 59.8, pp. 1170–1182.

Grundy, Scott M et al. (2005). "Diagnosis and management of the metabolic syndrome: an American Heart Association/National Heart, Lung, and Blood Institute scientific statement". In: *Circulation* 112.17, pp. 2735–2752.
DOI: 10.1161/circulationaha.105.169404.

Halkidi, Maria, Yannis Batistakis, and Michalis Vazirgiannis (2001). "On clustering validation techniques". In: *Journal of Intelligent Information Systems* 17.2-3, pp. 107–145.
DOI: 10.1023/A:1012801612483.

Hall, Zoe et al. (2017). "Lipid zonation and phospholipid remodeling in nonalcoholic fatty liver disease". In: *Hepatology* 65.4, pp. 1165–1180.
DOI: 10.1002/hep.28953.

Hamilton, Carlene A (1997). "Low-density lipoprotein and oxidised low-density lipoprotein: their role in the development of atherosclerosis". In: *Pharmacology & Therapeutics* 74.1, pp. 55–72.
DOI: 10.1016/S0163-7258(96)00202-1.

Harbison, Christopher T et al. (2004). "Transcriptional regulatory code of a eucaryotic genome". In: *Nature* 431.7004, pp. 99–104.
DOI: 10.1038/nature02800.

Hartigan, John A and Manchek A Wong (1979). "Algorithm AS 136: A *k*-means clustering algorithm". In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1, pp. 100–108.
DOI: 10.2307/2346830.

Hasin, Yehudit, Marcus Seldin, and Aldons Lusis (2017). "Multi-omics approaches to disease". In: *Genome Biology* 18.1, pp. 1–15.
DOI: 10.1186/s13059-017-1215-1.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.
URL: https://web.stanford.edu/~hastie/ElemStatLearn/.

Hastie, Trevor et al. (2019). *impute: Imputation for microarray data*. R package version 1.60.0.
DOI: 10.18129/B9.bioc.impute.

Hatzis, Christos et al. (2011). "A genomic predictor of response and survival following taxane-anthracycline chemotherapy for invasive breast cancer". In: *The Journal of the American Medical Association* 305.18, pp. 1873–1881.
DOI: 10.1001/jama.2011.593.

He, Xiaofei and Partha Niyogi (2004). "Locality preserving projections". In: *Advances in Neural Information Processing Systems*, pp. 153–160.
URL: http://papers.nips.cc/paper/2359-locality-preserving-projections.pdf.

Heller, Katherine A and Zoubin Ghahramani (2005). "Bayesian hierarchical clustering". In: *Proceedings of the 22nd International Conference on Machine Learning*. ACM, pp. 297–304.
DOI: 10.1145/1102351.1102389.

Hoadley, Katherine A et al. (2014). "Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin". In: *Cell* 158.4, pp. 929–944.
DOI: 10.1016/j.cell.2014.06.049.

Hoerl, Arthur E and Robert W Kennard (1970). "Ridge regression: Biased estimation for nonorthogonal problems". In: *Technometrics* 12.1, pp. 55–67.
DOI: 10.1080/00401706.1970.10488634.

Hollywood, Katherine, Daniel R Brison, and Royston Goodacre (2006). "Metabolomics: current technologies and future trends". In: *Proteomics* 6.17, pp. 4716–4723.
DOI: 10.1002/pmic.200600106.

Huang, Sijia, Kumardeep Chaudhary, and Lana X Garmire (2017). "More is better: recent progress in multi-omics data integration methods". In: *Frontiers in Genetics* 8, p. 84.
DOI: 10.3389/fgene.2017.00084.

Huang-Doran, Isabel et al. (2010). "Lipodystrophy: metabolic insights from a rare disorder." In: *The Journal of Endocrinology* 207.3, pp. 245–255.
DOI: 10.1677/joe-10-0272.

Hubert, Lawrence and Phipps Arabie (1985). "Comparing partitions". In: *Journal of Classification* 2.1, pp. 193–218.
DOI: 10.1007/BF01908075.

Ihmels, Jan et al. (2002). "Revealing modular organization in the yeast transcriptional network". In: *Nature Genetics* 31.4, p. 370.
DOI: 10.1038/ng941.

Ioannidis, John PA (2005). "Microarrays and molecular research: noise discovery?" In: *Lancet* 365.9458, p. 454.
DOI: 10.1016/S0140-6736(05)17878-7.

Jain, Sonia and Radford M Neal (2004). "A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model". In: *Journal of computational and Graphical Statistics* 13.1, pp. 158–182.
DOI: 10.1198/1061860043001.

James, Gareth et al. (2013). *An introduction to statistical learning*. Vol. 112. Springer.
DOI: 10.1007/978-1-4614-7138-7.

Jaworski, Piotr et al. (2010). *Copula theory and its applications*. Vol. 198. Springer.
DOI: 10.1007/978-3-642-12465-5.

Johnson, W Evan, Cheng Li, and Ariel Rabinovic (2007). "Adjusting batch effects in microarray expression data using empirical Bayes methods". In: *Biostatistics* 8.1, pp. 118–127.
DOI: 10.1093/biostatistics/kxj037.

Jones, Donald R, Matthias Schonlau, and William J Welch (1998). "Efficient global optimization of expensive black-box functions". In: *Journal of Global Optimization* 13.4, pp. 455–492.
DOI: 10.1023/A:1008306431147.

Joyce, Andrew R and Bernhard Ø Palsson (2006). "The model organism as a system: integrating 'omics' data sets". In: *Nature Reviews Molecular Cell Biology* 7.3, pp. 198–210.
DOI: 10.1038/nrm1857.

Kahn, Steven E, Rebecca L Hull, and Kristina M Utzschneider (2006). "Mechanisms linking obesity to insulin resistance and type 2 diabetes". In: *Nature* 444.7121, pp. 840–846.
DOI: 10.1038/nature05482.

Karczewski, Konrad J and Michael P Snyder (2018). "Integrative omics for health and disease". In: *Nature Reviews Genetics* 19.5, p. 299.
DOI: 10.1038/nrg.2018.4.

Katsuki, Akira et al. (2001). "Homeostasis model assessment is a reliable indicator of insulin resistance during follow-up of patients with type 2 diabetes". In: *Diabetes Care* 24.2, pp. 362–365.
DOI: 10.2337/diacare.24.2.362.

Kaufman, Leonard and Peter J Rousseeuw (1990). *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.
DOI: 10.1002/9780470316801.

Kieć-Klimczak, Małgorzata, Małgorzata Malczewska-Malec, and Bohdan Huszno (2008). "Leptin to adiponectin ratio, as an index of insulin resistance and atherosclerosis development". In: *Przeglad Lekarski* 65.12, pp. 844–849.

Kim, Seyoung, Eric P Xing, et al. (2012). "Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping". In: *The Annals of Applied Statistics* 6.3, pp. 1095–1117.
DOI: 10.1214/12-AOAS549.

Kim, Sinae, Mahlet G Tadesse, and Marina Vannucci (2006). "Variable selection in clustering via Dirichlet process mixture models". In: *Biometrika* 93.4, pp. 877–893.
DOI: 10.1093/biomet/93.4.877.

Kim, SungHwan et al. (2015). "Integrative phenotyping framework (iPF): integrative clustering of multiple omics data identifies novel lung disease subphenotypes". In: *BMC Genomics* 16.1, p. 924.
DOI: 10.1186/s12864-015-2170-4.

Kim, Sunghwan et al. (2017). "Integrative clustering of multi-level omics data for disease subtype discovery using sequential double regularization". In: *Biostatistics* 18.1, pp. 165–179.
DOI: 10.1093/biostatistics/kxw039.

Kirk, Erik P and Samuel Klein (2009). "Pathogenesis and pathophysiology of the cardiometabolic syndrome". In: *The Journal of Clinical Hypertension* 11.12, pp. 761–765.
DOI: 10.1111/j.1559-4572.2009.00054.x.

Kirk, Paul DW et al. (2012). "Bayesian correlated clustering to integrate multiple datasets". In: *Bioinformatics* 28.24, pp. 3290–3297.
DOI: 10.1093/bioinformatics/bts595.

Klein, Robert J et al. (2005). "Complement factor H polymorphism in age-related macular degeneration". In: *Science* 308.5720, pp. 385–389.
DOI: 10.1126/science.1109557.

Knerr, Stefan, Léon Personnaz, and Gérard Dreyfus (1990). "Single-layer learning revisited: a stepwise procedure for building and training a neural network". In: *Neurocomputing: Algorithms, Architectures and Applications* 68.41-50, p. 71.
DOI: 10.1007/978-3-642-76153-9_5.

Koestler, Devin C et al. (2010). "Semi-supervised recursively partitioned mixture models for identifying cancer subtypes". In: *Bioinformatics* 26.20, pp. 2578–2585.
DOI: 10.1093/bioinformatics/btq470.

Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". In: *International Joint Conference of Artificial Intelligence*. Vol. 14. 2, pp. 1137–1145.

Kolde, Raivo et al. (2012). "Robust rank aggregation for gene list integration and meta-analysis". In: *Bioinformatics* 28.4, pp. 573–580.
DOI: 10.1093/bioinformatics/btr709.

Koning, AP Jason de et al. (2011). "Repetitive elements may comprise over two-thirds of the human genome". In: *PLoS Genetics* 7.12, e1002384.
DOI: 10.1371/journal.pgen.1002384.

Kristensen, Vessela N et al. (2012). "Integrated molecular profiles of invasive breast tumors and ductal carcinoma in situ (DCIS) reveal differential vascular and interleukin signaling". In: *Proceedings of the National Academy of Sciences* 109.8, pp. 2802–2807.
DOI: 10.1073/pnas.1108781108.

Kristensen, Vessela N et al. (2014). "Principles and methods of integrative genomic analyses in cancer." In: *Nature Reviews Cancer* 14.5, pp. 299–313.
DOI: 10.1038/nrc3721.

Kulis, Brian et al. (2013). "Metric learning: A survey". In: *Foundations and Trends® in Machine Learning* 5.4, pp. 287–364.
DOI: 10.1561/2200000019.

Kuo, Min-Hao and C David Allis (1998). "Roles of histone acetyltransferases and deacetylases in gene regulation". In: *Bioessays* 20.8, pp. 615–626.
DOI: 10.1002/(SICI)1521-1878(199808)20:8<615::AID-BIES4>3.0.CO;2-H.

Lala, Vasimahmed and David A Minter (2018). "Liver function tests". In: *StatPearls*. StatPearls Publishing.
URL: http://europepmc.org/books/NBK482489.

Lanckriet, Gert RG et al. (2004a). "A statistical framework for genomic data fusion". In: *Bioinformatics* 20.16, pp. 2626–2635.
DOI: 10.1093/bioinformatics/bth294.

– (2004b). "Learning the kernel matrix with semidefinite programming". In: *Journal of Machine Learning Research* 5.Jan, pp. 27–72.
URL: https://www.jmlr.org/papers/v5/lanckriet04a.html.

Latchman, David S (1997). "Transcription factors: an overview". In: *The international journal of biochemistry & cell biology* 29.12, pp. 1305–1312.
DOI: 10.1016/S1357-2725(97)00085-X.

Lavit, Christine et al. (1994). "The ACT (STATIS method)". In: *Computational Statistics & Data Analysis* 18.1, pp. 97–119.
DOI: 10.1016/0167-9473(94)90134-1.

Lee, James C et al. (2011). "Gene expression profiling of CD8+ T cells predicts prognosis in patients with Crohn disease and ulcerative colitis". In: *The Journal of clinical investigation* 121.10, pp. 4170–4179.
DOI: 10.1172/JCI59255.

Leek, Jeffrey T et al. (2010). "Tackling the widespread and critical impact of batch effects in high-throughput data". In: *Nature Reviews Genetics* 11.10, pp. 733–739.
DOI: 10.1038/nrg2825.

Leek, Jeffrey T. et al. (2019). *sva: Surrogate Variable Analysis*. R package version 3.34.0.
DOI: 10.18129/B9.bioc.sva.

Lemsara, Amina, Salima Ouadfel, and Holger Fröhlich (2020). "PathME: pathway based multi-modal sparse autoencoders for clustering of patient-level multi-omics data". In: *BMC Bioinformatics* 21.1, pp. 1–20.
DOI: 10.1186/s12859-020-3465-2.

Levine, Adrian B et al. (2019). "Rise of the machines: Advances in deep learning for cancer diagnosis". In: *Trends in Cancer*.
DOI: 10.1016/j.trecan.2019.02.002.

Li, Xue, Xinlei Wang, and Guanghua Xiao (2019). "A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications". In: *Briefings in Bioinformatics* 20.1, pp. 178–189.
DOI: 10.1093/bib/bbx101.

Li, Yifeng, Fang-Xiang Wu, and Alioune Ngom (2018). "A review on machine learning principles for multi-view biological data integration". In: *Briefings in Bioinformatics* 19.2, pp. 325–340.
DOI: 10.1093/bib/bbw113.

Liang, Muxuan et al. (2014). "Integrative data analysis of multi-platform cancer data with a multimodal deep learning approach". In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12.4, pp. 928–937.
DOI: 10.1109/TCBB.2014.2377729.

Lin, Shili (2010). "Rank aggregation methods". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.5, pp. 555–570.
DOI: 10.1002/wics.111.

Lin, Shili and Jie Ding (2009). "Integration of ranked lists via cross entropy Monte Carlo with applications to mRNA and microRNA studies". In: *Biometrics* 65.1, pp. 9–18.
DOI: 10.1111/j.1541-0420.2008.01044.x.

Lindsay, Tim et al. (2019). "Descriptive epidemiology of physical activity energy expenditure in UK adults (The Fenland study)". In: *International Journal of Behavioral Nutrition and Physical Activity* 16.1, p. 126.
DOI: 10.1186/s12966-019-0882-6.

Liu, Jie et al. (2018). "Data integration by multi-tuning parameter elastic net regression". In: *BMC Bioinformatics* 19.1, pp. 1–9.
DOI: 10.1186/s12859-018-2401-1.

Liu, Xiangdong et al. (2007). "Bayesian hierarchical model for transcriptional module discovery by jointly modeling gene expression and ChIP-chip data". In: *BMC Bioinformatics* 8.1, p. 283.
DOI: 10.1186/1471-2105-8-283.

Liverani, Silvia et al. (2015). "PReMiuM: An R package for profile regression mixture models using Dirichlet processes". In: *Journal of Sstatistical Software* 64.7, p. 1.
DOI: 10.18637/jss.v064.i07.

Lock, Eric F and David B Dunson (2013). "Bayesian consensus clustering". In: *Bioinformatics*, btt425.
DOI: 10.1093/bioinformatics/btt425.

Lock, Eric F et al. (2013). "Joint and individual variation explained (JIVE) for integrated analysis of multiple data types". In: *The annals of applied statistics* 7.1, p. 523.

Luenberger, David G and Yinyu Ye (1984). *Linear and nonlinear programming*. Springer.
DOI: 10.1007/978-3-319-18842-3.

Lunn, David et al. (2012). *The BUGS book: A practical introduction to Bayesian analysis*. CRC press.

Ma, Tianle and Aidong Zhang (2018). "Affinity network fusion and semi-supervised learning for cancer patient clustering". In: *Methods* 145, pp. 16–24.
DOI: 10.1016/j.ymeth.2018.05.020.

Mann, Henry B and Donald R Whitney (1947). "On a test of whether one of two random variables is stochastically larger than the other". In: *The Annals of Mathematical Statistics*, pp. 50–60.
DOI: 10.1214/aoms/1177730491.

Mariette, Jérôme and Nathalie Villa-Vialaneix (2018). "Unsupervised multiple kernel learning for heterogeneous data integration". In: *Bioinformatics* 34.6, pp. 1009–1015.
DOI: 10.1093/bioinformatics/btx682.

Marshall, Eliot (2004). "Getting the noise out of gene arrays". In: *Science* 306.5696, p. 630.
URL: https://ezp.lib.cam.ac.uk/login?url=https://search.proquest.com/docview/213572572?accountid=9851.

Mason, Samuel A et al. (2016). "MDI-GPU: accelerating integrative modelling for genomic-scale data using GP-GPU computing." In: *Statistical Applications in Genetics and Molecular Biology* 15.1, pp. 83–86.
DOI: 10.1515/sagmb-2015-0055.

McCarroll, Steven A and David M Altshuler (2007). "Copy-number variation and association studies of human disease". In: *Nature genetics* 39.7, S37–S42.
DOI: 10.1038/ng2080.

McIntyre, Lauren M et al. (2000). "Circumventing multiple testing: a multilocus Monte Carlo approach to testing for association". In: *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society* 19.1, pp. 18–29.
DOI: 10.1002/1098-2272(200007)19:1<18::AID-GEPI2>3.0.CO;2-Y.

McKay, Michael D, Richard J Beckman, and William J Conover (1979). "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code". In: *Technometrics* 21.2, pp. 239–245.
DOI: 10.1080/00401706.1979.10489755.

McLachlan, Geoffrey J and David Peel (2004). *Finite mixture models*. John Wiley & Sons.

Medvedovic, Mario and Siva Sivaganesan (2002). "Bayesian infinite mixture model based clustering of gene expression profiles". In: *Bioinformatics* 18.9, pp. 1194–1206.
DOI: 10.1093/bioinformatics/18.9.1194.

Meilă, Marina (2007). "Comparing clusterings—an information based distance". In: *Journal of Multivariate Analysis* 98.5, pp. 873–895.
DOI: 10.1016/j.jmva.2006.11.013.

Mika, Sebastian et al. (1999). "Fisher discriminant analysis with kernels". In: *Proceedings of the 1999 IEEE Signal Processing Society Workshop.* IEEE, pp. 41–48.
DOI: 10.1109/NNSP.1999.788121.

Milligan, Glenn W and Martha C Cooper (1985). "An examination of procedures for determining the number of clusters in a data set". In: *Psychometrika* 50.2, pp. 159–179.
DOI: 10.1007/BF02294245.

Mistry, Meeta and Paul Pavlidis (2008). "Gene Ontology term overlap as a measure of gene functional similarity." In: *BMC Bioinformatics* 9, p. 327.
DOI: 10.1186/1471-2105-9-327.

Mo, Qianxing et al. (2013). "Pattern discovery and cancer gene identification in integrated cancer genomic data". In: *Proceedings of the National Academy of Sciences* 110.11, pp. 4245–4250.
DOI: 10.1073/pnas.1208949110.

Mo, Qianxing et al. (2018). "A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data". In: *Biostatistics* 19.1, pp. 71–86.
DOI: 10.1093/biostatistics/kxx017.

Molitor, John et al. (2010). "Bayesian profile regression with an application to the National Survey of Children's Health". In: *Biostatistics* 11.3, pp. 484–498.
DOI: 10.1093/biostatistics/kxq013.

Monti, Stefano et al. (2003). "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data". In: *Machine Learning* 52.1-2, pp. 91–118.
DOI: 10.1023/A:1023949509487.

Moons, Karel GM et al. (2015). "Transparent Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis (TRIPOD): explanation and elaboration". In: *Annals of Internal Medicine* 162.1, W1–W73.
DOI: 10.7326/m14-0698.

Moore, Lisa D, Thuc Le, and Guoping Fan (2013). "DNA methylation and its basic function". In: *Neuropsychopharmacology* 38.1, pp. 23–38.
DOI: 10.1038/npp.2012.112.

Moore, Steven C et al. (2014). "Human metabolic correlates of body mass index". In: *Metabolomics* 10.2, pp. 259–269.
DOI: 10.1007/s11306-013-0574-1.

Murphy, Kevin P (2012). *Machine learning: a probabilistic perspective*. MIT press.
URL: https://www.cs.ubc.ca/~murphyk/MLbook/.

Neal, Radford M (2000). "Markov Chain Sampling Methods for Dirichlet Process Mixture Models". In: *Journal of Computational and Graphical Statistics* 9.2, pp. 249–265.
DOI: 10.1080/10618600.2000.10474879.

Nelsen, Roger B (2007). *An introduction to copulas*. Springer Science & Business Media.
DOI: 10.1007/0-387-28678-0.

Ng, Andrew Y, Michael I Jordan, and Yair Weiss (2002). "On spectral clustering: Analysis and an algorithm". In: *Advances in Neural Information Processing Systems*, pp. 849–856.
URL: http://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf.

Nguyen, Hung et al. (2019). "PINSPlus: a tool for tumor subtype discovery in integrated genomic data". In: *Bioinformatics* 35.16, pp. 2843–2846.
DOI: 10.1093/bioinformatics/bty1049.

Nguyen, Tin et al. (2017). "A novel approach for data integration and disease subtyping". In: *Genome Research* 27.12, pp. 2025–2039.
DOI: 10.1101/gr.215129.116.

Ni, Yang et al. (2020). "Scalable bayesian nonparametric clustering and classification". In: *Journal of Computational and Graphical Statistics* 29.1, pp. 53–65.
DOI: 10.1080/10618600.2019.1624366.

Nicora, Giovanna et al. (2020). "Integrated Multi-Omics Analyses in Oncology: A Review of Machine Learning Methods and Tools". In: *Frontiers in Oncology* 10, p. 1030.
DOI: 10.3389/fonc.2020.01030.

Nordestgaard, Børge G et al. (2007). "Nonfasting triglycerides and risk of myocardial infarction, ischemic heart disease, and death in men and women". In: *The Journal of the American Medical Association* 298.3, pp. 299–308.
DOI: 10.1001/jama.298.3.299.

Oh, Deborah K, Theodore Ciaraldi, and Robert R Henry (2007). "Adiponectin in health and disease". In: *Diabetes, Obesity and Metabolism* 9.3, pp. 282–289.
DOI: 10.1111/j.1463-1326.2006.00610.x.

Ozsolak, Fatih and Patrice M Milos (2011). "RNA sequencing: advances, challenges and opportunities". In: *Nature Reviews Genetics* 12.2, pp. 87–98.
DOI: 10.1038/nrg2934.

Papathomas, Michail et al. (2011). "Examining the joint effect of multiple risk factors using exposure risk profiles: lung cancer in nonsmokers". In: *Environmental health perspectives* 119.1, pp. 84–91.
DOI: 10.1289/ehp.1002118.

Platt, John C (1999). "Fast training of support vector machines using sequential minimal optimization". In: *Advances in Kernel Methods – Support Vector Learning*. Ed. by Bernhard Schölkopf, Christopher JC Burges, Alexander J Smola. MIT Press. Chap. 12, pp. 185–208.
DOI: 10.1109/ISKE.2008.4731075.

Rakotomamonjy, Alain and Francis Bach (2007). "More efficiency in multiple kernel learning". In: *Proceedings of the 24th International Conference on Machine Learning*, pp. 775–782.
DOI: 10.1145/1273496.1273594.

Rakotomamonjy, Alain et al. (2008). "SimpleMKL". In: *Journal of Machine Learning Research* 9.Nov, pp. 2491–2521.
URL: http://www.jmlr.org/papers/v9/rakotomamonjy08a.html.

Ramazzotti, Daniele et al. (2018). "Multi-omic tumor data reveal diversity of molecular mechanisms that correlate with survival". In: *Nature Communications* 9.1, pp. 1–14.
DOI: 10.1038/s41467-018-06921-8.

Rand, William M. (1971). "Objective Criteria for the Evaluation of Clustering Methods". In: *Journal of the American Statistical Association* 66.336, pp. 846–850.
DOI: 10.1080/01621459.1971.10482356.

Rappoport, Nimrod, Roy Safra, and Ron Shamir (2020). "MONET: Multi-omic patient module detection by omic selection". In: *bioRxiv*.
DOI: 10.1101/2020.02.21.960062.

Rappoport, Nimrod and Ron Shamir (2018). "Multi-omic and multi-view clustering algorithms: review and cancer benchmark". In: *Nucleic Acids Research* 46.20, pp. 10546–10562.
DOI: 10.1093/nar/gky889.

– (2019). "NEMO: Cancer subtyping by integration of partial multi-omic data". In: *Bioinformatics* 35.18, pp. 3348–3356.
DOI: 10.1093/bioinformatics/btz058.

Rasmussen, Carl Edward (2000). "The infinite Gaussian mixture model". In: *Advances in Neural Information Processing Systems*, pp. 554–560.
URL: http://papers.nips.cc/paper/1745-the-infinite-gaussian-mixture-model.pdf.

Rasmussen, Carl Edward and Christopher K Williams (2006). *Gaussian processes for machine learning*. Vol. 2. 3. MIT Press.
URL: http://www.gaussianprocess.org/gpml/.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.
URL: https://www.r-project.org/.

Reaven, Gerald M et al. (1988). "Measurement of plasma glucose, free fatty acid, lactate, and insulin for 24h in patients with NIDDM". In: *Diabetes* 37.8, pp. 1020–1024.
DOI: 10.2337/diab.37.8.1020.

Reynolds, C Patrick, Barry J Maurer, and Richard N Kolesnick (2004). "Ceramide synthesis and metabolism as a target for cancer therapy". In: *Cancer letters* 206.2, pp. 169–180.
DOI: 10.1016/j.canlet.2003.08.034.

Richardson, Sylvia and Peter Green (1997). "On Bayesian Analysis of Mixtures with an Unknown Number of Components". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59.4, pp. 731–792.
DOI: 10.1111/1467-9868.00095.

Richardson, Sylvia, George C Tseng, and Wei Sun (2016). "Statistical methods in integrative genomics". In: *Annual Reviews of Statistics and Its Applications*.
DOI: 10.1146/annurev-statistics-041715-033506.

Ridker, Paul M (2001). "High-sensitivity C-reactive protein: potential adjunct for global risk assessment in the primary prevention of cardiovascular disease". In: *Circulation* 103.13, pp. 1813–1818.
DOI: 10.1161/01.CIR.103.13.1813.

Riley, Richard D et al. (2020). "Calculating the sample size required for developing a clinical prediction model". In: *BMJ* 368.
DOI: 10.1136/bmj.m441.

Röder, Benedict et al. (2019). "web-rMKL: a web server for dimensionality reduction and sample clustering of multi-view data based on unsupervised multiple kernel learning". In: *Nucleic Acids Research* 47.W1, W605–W609.
DOI: 10.1093/nar/gkz422.

Rogers, Simon and Mark Girolami (2016). *A first course in machine learning*. CRC Press.
URL: http://www.dcs.gla.ac.uk/~srogers/firstcourseml/.

Rogers, Simon et al. (2008). "Investigating the correspondence between transcriptomic and proteomic expression profiles using coupled cluster models". In: *Bioinformatics* 24.24, pp. 2894–2900.
DOI: 10.1093/bioinformatics/btn553.

Rohart, Florian et al. (2017). "mixOmics: An R package for 'omics feature selection and multiple data integration". In: *PLoS computational biology* 13.11, e1005752.
DOI: 10.1371/journal.pcbi.1005752.

Roth, Volker and Volker Steinhage (2000). "Nonlinear discriminant analysis using kernel functions". In: *Advances in Neural Information Processing Systems*, pp. 568–574.
URL: https://dl.acm.org/doi/abs/10.5555/3009657.3009738.

Rousseau, Judith and Kerrie Mengersen (2011). "Asymptotic behaviour of the posterior distribution in overfitted mixture models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73.5, pp. 689–710.
DOI: 10.1111/j.1467-9868.2011.00781.x.

Rousseeuw, Peter J. (1987). "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". In: *Journal of Computational and Applied Mathematics* 20.C, pp. 53–65.
DOI: 10.1016/0377-0427(87)90125-7.

Sanders, Francis WB et al. (2018). "Hepatic steatosis risk is partly driven by increased de novo lipogenesis following carbohydrate consumption". In: *Genome Biology* 19.1, p. 79.
DOI: 10.1186/s13059-018-1439-8.

Sarwar, Nadeem et al. (2007). "Triglycerides and the risk of coronary heart disease: 10,158 incident cases among 262,525 participants in 29 Western prospective studies". In: *Circulation* 115.4, 450—458.
DOI: 10.1161/circulationaha.106.637793.

Sato, Yusuke et al. (2013). "Integrated molecular analysis of clear-cell renal cell carcinoma". In: *Nature Genetics* 45.8, pp. 860–867.
DOI: 10.1038/nature10166.

Savage, Richard S et al. (2010). "Discovering transcriptional modules by Bayesian data integration". In: *Bioinformatics* 26.12, pp. i158–i167.
DOI: 10.1093/bioinformatics/btq210.

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller (1998). "Nonlinear component analysis as a kernel eigenvalue problem". In: *Neural Computation* 10.5, pp. 1299–1319.
DOI: 10.1162/089976698300017467.

Schölkopf, Bernhard and Alexander J Smola (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Sebat, Jonathan et al. (2007). "Strong association of de novo copy number mutations with autism". In: *Science* 316.5823, pp. 445–449.
DOI: 10.1126/science.1138659.

Şenbabaoğlu, Yasin, George Michailidis, and Jun Z. Li (2014). "Critical limitations of consensus clustering in class discovery". In: *Scientific Reports* 4.6207.
DOI: 10.1038/srep06207.

Seoane, José A et al. (2014). "A pathway-based data integration framework for prediction of disease progression". In: *Bioinformatics* 30.6, pp. 838–845.
DOI: 10.1093/bioinformatics/btt610.

Sethuraman, Jayaram (1994). "A constructive definition of Dirichlet priors". In: *Statistica Sinica* 4.2, pp. 639–650.
URL: https://www.jstor.org/stable/24305538.

Seyres, Denis et al. (2020). "Transcriptional, epigenetic and metabolic signatures in cardiometabolic syndrome defined by extreme phenotypes". In: *bioRxiv*.
DOI: 10.1101/2020.03.06.961805.

Shapiro, James A (2009). "Revisiting the central dogma in the 21st century". In: *Annals of the New York Academy of Sciences* 1178.1, pp. 6–28.
DOI: 10.1111/j.1749-6632.2009.04990.x.

Shawe-Taylor, John and Nello Cristianini (2004). *Kernel methods for pattern analysis*. Cambridge University Press.
DOI: 10.1017/CBO9780511809682.

Shen, Ronglai (2012). *iCluster: Integrative clustering of multiple genomic data types*. R package version 2.1.0.
URL: https://CRAN.R-project.org/package=iCluster.

Shen, Ronglai, Adam B Olshen, and Marc Ladanyi (2009). "Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis". In: *Bioinformatics* 25.22, pp. 2906–2912.
DOI: 10.1093/bioinformatics/btp543.

Shen, Ronglai, Sijian Wang, and Qianxing Mo (2013). "Sparse integrative clustering of multiple omics data sets". In: *The Annals of Applied Statistics* 7.1, p. 269.
DOI: 10.1214/12-AOAS578.

Shen, Ronglai et al. (2012). "Integrative subtype discovery in glioblastoma using iCluster". In: *PloS One* 7.4.
DOI: 10.1371/journal.pone.0035236.

Shi, Yuguang and Paul Burn (2004). "Lipid metabolic enzymes: emerging drug targets for the treatment of obesity". In: *Nature Reviews Drug discovery* 3.8, pp. 695–710.
DOI: 10.1038/nrd1469.

Sill, Martin et al. (2014). "c060: Extended inference with lasso and elastic-net regularized Cox and generalized linear models". In: *Journal of Statistical Software* 62.5, pp. 1–22.
DOI: 10.18637/jss.v062.i05.

Simidjievski, Nikola et al. (2019). "Variational autoencoders for cancer data integration: design principles and computational practice". In: *Frontiers in Genetics* 10.1205.
DOI: 10.3389/fgene.2019.01205.

Sirtori, Cesare R (2006). "HDL and the progression of atherosclerosis: new insights". In: *European Heart Journal Supplements* 8.suppl_F, F4–F9.
DOI: 10.1093/eurheartj/sul034.

Sokal, Robert R and F James Rohlf (1962). "The comparison of dendrograms by objective methods". In: *Taxon* 11.2, pp. 33–40.
DOI: 10.2307/1217208.

Søndergaard, Esben et al. (2017). "How to measure adipose tissue insulin sensitivity". In: *The Journal of Clinical Endocrinology & Metabolism* 102.4, pp. 1193–1199.
DOI: 10.1210/jc.2017-00047.

Speicher, Nora K and Nico Pfeifer (2015). "Integrating different data types by regularized unsupervised multiple kernel learning with application to cancer subtype discovery". In: *Bioinformatics* 31.12, pp. i268–i275.
DOI: 10.1093/bioinformatics/btv244.

St Clair, David (2009). "Copy number variation and schizophrenia". In: *Schizophrenia bulletin* 35.1, pp. 9–12.
DOI: 10.1093/schbul/sbn147.

Steinhaus, Hugo (1956). "Sur la division des corps matériels en parties". In: *Bulletin de l'Académie Polonaise des Sciences* IV.12, pp. 801–804.

Storey, John D. et al. (2019). *qvalue: Q-value estimation for false discovery rate control*. R package version 2.18.0.
DOI: 10.18129/B9.bioc.qvalue.

Strauß, Magdalena E et al. (2020). "GPseudoClust: deconvolution of shared pseudoprofiles at single-cell resolution". In: *Bioinformatics* 36.5, pp. 1484–1491.
DOI: 10.1093/bioinformatics/btz778.

Strehl, Alexander and Joydeep Ghosh (2002). "Cluster ensembles—a knowledge reuse framework for combining multiple partitions". In: *Journal of Machine Learning Research* 3.Dec, pp. 583–617.
URL: http://www.jmlr.org/papers/v3/strehl02a.html.

Subramanian, Indhupriya et al. (2020). "Multi-omics data integration, interpretation, and its application". In: *Bioinformatics and Biology Insights* 14, pp. 1–24.
DOI: 10.1177/1177932219899051.

Tadesse, Mahlet G, Naijun Sha, and Marina Vannucci (2005). "Bayesian variable selection in clustering high-dimensional data". In: *Journal of the American Statistical Association* 100.470, pp. 602–617.
DOI: 10.1198/016214504000001565.

Talayero, Beatriz G and Frank M Sacks (2011). "The role of triglycerides in atherosclerosis". In: *Current Cardiology Reports* 13.6, p. 544.
DOI: 10.1007/s11886-011-0220-3.

Tam, Vivian et al. (2019). "Benefits and limitations of genome-wide association studies". In: *Nature Reviews Genetics* 20.8, pp. 467–484.
DOI: 10.1038/s41576-019-0127-1.

The Cancer Genome Atlas Research Network (2012). "Comprehensive molecular portraits of human breast tumours." In: *Nature* 487.7407, pp. 61–70.
DOI: 10.1038/nature11412.

– (2013). "Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia". In: *New England Journal of Medicine* 368.22, pp. 2059–2074.
DOI: 10.1056/NEJMoa1301689.
– (2017). "Comprehensive and integrative genomic characterization of hepatocellular carcinoma". In: *Cell* 169.7, pp. 1327–1341.
DOI: 10.1016/j.cell.2017.05.046.

The Cancer Genome Atlas Research Network et al. (2011). "Integrated genomic analyses of ovarian carcinoma". In: *Nature* 474.7353, p. 609.
DOI: 10.1038/nature10166.

The Cancer Genome Atlas Research Network et al. (2013). "The cancer genome atlas pan-cancer analysis project". In: *Nature Genetics* 45.10, p. 1113.
DOI: 10.1038/ng.2764.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.
DOI: 10.1111/j.2517-6161.1996.tb02080.x.

Tibshirani, Robert et al. (2001). "Estimating the number of clusters in a data set via the gap statistic". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63.2, pp. 411–423.
DOI: 10.1111/1467-9868.00293.

Tibshirani, Robert et al. (2005). "Sparsity and smoothness via the fused lasso". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.1, pp. 91–108.
DOI: 10.1111/j.1467-9868.2005.00490.x.

Topchy, Alexander, Anil K Jain, and William Punch (2004). "A mixture model for clustering ensembles". In: *Proceedings of the 2004 SIAM International Conference on Data Mining*. SIAM, pp. 379–390.
DOI: 10.1137/1.9781611972740.35.

Trivedi, Pravin K, David M Zimmer, et al. (2007). "Copula modeling: an introduction for practitioners". In: *Foundations and Trends® in Econometrics* 1.1, pp. 1–111.
DOI: 10.1561/0800000005.

Troyanskaya, Olga et al. (2001). "Missing value estimation methods for DNA microarrays". In: *Bioinformatics* 17.6, pp. 520–525.
DOI: 10.1093/bioinformatics/17.6.520.

Tweeddale, Helen, Lucinda Notley-McRobb, and Thomas Ferenci (1998). "Effect of slow growth on metabolism of Escherichia coli, as revealed by global metabolite pool ("metabolome") analysis". In: *Journal of bacteriology* 180.19, pp. 5109–5116.
DOI: 10.1128/JB.180.19.5109-5116.1998.

van Buuren, Stef and Karin Groothuis-Oudshoorn (2011). "mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45.3, pp. 1–67.
DOI: 10.18637/jss.v045.i03.

van Tuijl, Julia et al. (2019). "Immunometabolism orchestrates training of innate immunity in atherosclerosis". In: *Cardiovascular Research* 115.9, pp. 1416–1424.
DOI: 10.1093/cvr/cvz107.

Vapnik, Vladimir N (1999). "An overview of statistical learning theory". In: *IEEE Transactions on Neural Networks* 10.5, pp. 988–999.
DOI: 10.1109/72.788640.

Vats, Dootika and Christina Knudson (2018). *Revisiting the Gelman-Rubin diagnostic*.
arXiv: 1812.09384.

Velten, Britta and Wolfgang Huber (2018). *Adaptive penalization in high-dimensional regression and classification with external covariates using variational Bayes*.
arXiv: 1811.02962.

Venes, Donald (2017). *Taber's cyclopedic medical dictionary*. FA Davis Company.

Venkitaraman, Ashok R (2014). "Cancer suppression by the chromosome custodians, BRCA1 and BRCA2". In: *Science* 343.6178, pp. 1470–1475.
DOI: 10.1126/science.1252230.

Visscher, Peter M et al. (2012). "Five years of GWAS discovery". In: *The American Journal of Human Genetics* 90.1, pp. 7–24.
DOI: 10.1016/j.ajhg.2011.11.029.

Vlachos, Andreas, Anna Korhonen, and Zoubin Ghahramani (2009). "Unsupervised and constrained Dirichlet process mixture models for verb clustering". In: *Proceedings of the workshop on geometrical models of natural language semantics*, pp. 74–82.
URL: https://dl.acm.org/doi/abs/10.5555/1705415.1705425.

Wade, Sara, Zoubin Ghahramani, et al. (2018). "Bayesian cluster analysis: Point estimation and credible balls (with discussion)". In: *Bayesian Analysis* 13.2, pp. 559–626.
DOI: 10.1214/17-BA1073.

Wandall, Hans H et al. (2017). *Essentials of glycobiology*.

Wang, Bo et al. (2014). "Similarity network fusion for aggregating data types on a genomic scale". In: *Nature Methods* 11.3, p. 333.
DOI: 10.1038/nmeth.2810.

Wang, Bo et al. (2017). "Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning". In: *Nature Methods* 14, pp. 414–416.
DOI: 10.1038/nmeth.4207.

Wang, Bo et al. (2018). "SIMLR: A Tool for Large-Scale Genomic Analyses by Multi-Kernel Learning". In: *Proteomics* 18.2, p. 1700232.
DOI: 10.1002/pmic.201700232.

Wang, Ketong and Michael D Porter (2018). "Optimal Bayesian clustering using non-negative matrix factorization". In: *Computational Statistics & Data Analysis* 128, pp. 395–411.
DOI: 10.1016/j.csda.2018.08.002.

Wang, Lianming and David B Dunson (2011). "Fast Bayesian inference in Dirichlet process mixture models". In: *Journal of Computational and Graphical Statistics* 20.1, pp. 196–216.
DOI: 10.1198/jcgs.2010.07081.

Wang, Wenting et al. (2013). "iBAG: integrative Bayesian analysis of high-dimensional multiplatform genomics data". In: *Bioinformatics* 29.2, pp. 149–159.
DOI: 10.1093/bioinformatics/bts655.

Wasserman, David H (2009). "Four grams of glucose". In: *American Journal of Physiology-Endocrinology and Metabolism* 296.1, E11–E21.
DOI: 10.1152/ajpendo.90563.2008.

Watson, James et al. (2014). *Molecular biology of the gene*. Pearson Education Inc.

Wenk, Markus R (2005). "The emerging field of lipidomics". In: *Nature reviews Drug discovery* 4.7, pp. 594–610.
DOI: 10.1038/nrd1776.

Wilkerson, Matthew D and D Neil Hayes (2010). "ConsensusClusterPlus: A class discovery tool with confidence assessments and item tracking". In: *Bioinformatics* 26.12, pp. 1572–1573.
DOI: 10.1093/bioinformatics/btq170.

Wishart, DS, D Tzur, C Knox, et al. (2007). "HMDB: the Human Metabolome Database". In: *Nucleic Acids Research* 35.suppl 1, pp. D521–D526.
DOI: 10.1093/nar/gkl923.

Witten, Daniela M and Robert Tibshirani (2010). "A framework for feature selection in clustering". In: *Journal of the American Statistical Association* 105.490, pp. 713–726.
DOI: 10.1198/jasa.2010.tm09415.

– (2018). *sparcl: Perform Sparse Hierarchical Clustering and Sparse K-Means Clustering*. R package version 1.0.4.
URL: https://CRAN.R-project.org/package=sparcl.

World Health Organization. *WHO definitions of genetics and genomics*.
URL: https://www.who.int/genomics/geneticsVSgenomics/en/.

– *WHO key facts about cardiovascular diseases*.
URL: https://www.who.int/cardiovascular_diseases/about_cvd/en/.

Wright, Helen L et al. (2010). "Neutrophil function in inflammation and inflammatory diseases". In: *Rheumatology* 49.9, pp. 1618–1631.
DOI: 10.1093/rheumatology/keq045.

Wu, Dingming et al. (2015). "Fast dimension reduction and integrative clustering of multi-omics data using low-rank approximation: application to cancer molecular classification". In: *BMC Genomics* 16.1, p. 1022.
DOI: 10.1186/s12864-015-2223-8.

Wu, Lani F et al. (2002). "Large-scale prediction of Saccharomyces cerevisiae gene function using overlapping transcriptional clusters". In: *Nature genetics* 31.3, pp. 255–265.
DOI: 10.1038/ng906.

Xing, Eric P et al. (2003). "Distance metric learning with application to clustering with side-information". In: *Advances in Neural Information Processing Systems*, pp. 521–528.
URL: http://papers.nips.cc/paper/2164-distance-metric-learning-with-application-to-clustering-with-side-information.pdf.

Yan, Shuicheng et al. (2006). "Graph embedding and extensions: A general framework for dimensionality reduction". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.1, pp. 40–51.
DOI: 10.1109/TPAMI.2007.250598.

Yeung, Ka Yee, David R Haynor, and Walter L Ruzzo (2001). "Validating clustering for gene expression data". In: *Bioinformatics* 17.4, pp. 309–318.
DOI: 10.1093/bioinformatics/17.4.309.

Yong, Wai-Shin, Fei-Man Hsu, and Pao-Yang Chen (2016). "Profiling genome-wide DNA methylation". In: *Epigenetics & Chromatin* 9.1, p. 26.
DOI: 10.1186/s13072-016-0075-3.

Yu, Shi et al. (2010). "L2-norm multiple kernel learning and its application to biomedical data fusion". In: *BMC Bioinformatics* 11.
DOI: 10.1186/1471-2105-11-309.

Yu, Shipeng et al. (2006). "Supervised probabilistic principal component analysis". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 464–473.
DOI: 10.1145/1150402.1150454.

Yuan, Yinyin, Richard S Savage, and Florian Markowetz (2011). "Patient-specific data fusion defines prognostic cancer subtypes". In: *PLoS Computational Biology* 7.10.
DOI: 10.1371/journal.pcbi.1002227.

Zhang, Shihua et al. (2012). "Discovery of multi-dimensional modules by integrative analysis of cancer genomic data". In: *Nucleic Acids Research* 40.19, pp. 9379–9391.
DOI: 10.1093/nar/gks725.

Zhao, Qing et al. (2015). "Combining multidimensional genomic measurements for predicting cancer prognosis: observations from TCGA". In: *Briefings in Bioinformatics* 16.2, pp. 291–303.
DOI: 10.1093/bib/bbu003.

Zhao, Zhi and Manuela Zucknick (2020). "Structured penalized regression for drug sensitivity prediction". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 69.3.
DOI: 10.1111/rssc.12400.

Zhu, Zhiwei et al. (2012). "A multi-omic map of the lipid-producing yeast Rhodosporidium toruloides". In: *Nature communications* 3.1, pp. 1–12.
DOI: 10.1038/ncomms2112.

Zou, Hui and Trevor Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, pp. 301–320.
DOI: 10.1111/j.1467-9868.2005.00503.x.

Žurauskienė, Justina, Paul DW Kirk, and Michael PH Stumpf (2016). "A graph theoretical approach to data fusion". In: *Statistical Applications in Genetics and Molecular Biology* 15.2, pp. 107–122.
DOI: 10.1515/sagmb-2016-0016.